



A sequential approach for a multi-commodity two-echelon distribution problem

Wenjuan Gu, Claudia Archetti, Diego Cattaruzza, Maxime Ogier, Frédéric Semet, M Grazia Speranza

► To cite this version:

Wenjuan Gu, Claudia Archetti, Diego Cattaruzza, Maxime Ogier, Frédéric Semet, et al.. A sequential approach for a multi-commodity two-echelon distribution problem. *Computers & Industrial Engineering*, 2022, 163, pp.107793. hal-03167379

HAL Id: hal-03167379

<https://hal.science/hal-03167379>

Submitted on 12 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A sequential approach for a multi-commodity two-echelon distribution problem

Wenjuan Gu¹, Claudia Archetti², Diego Cattaruzza³, Maxime Ogier³, Frédéric Semet³, M.Grazia Speranza⁴

¹: School of Modern Posts, Beijing University of Posts and Telecommunications
100876 Beijing, China
wenjuan.gu@bupt.edu.cn

²: Department of Information Systems, Decision Sciences and Statistics
ESSEC Business School in Paris
95021 Cergy-Pontoise Cedex, France
archetti@essec.edu

³: Univ. Lille, CNRS, Centrale Lille, Inria
UMR 9189 - CRISTAL
Centre de Recherche en Informatique Signal et Automatique de Lille
F-59000 Lille, France
{diego.cattaruzza, maxime.ogier, frederic.semet}@centralelille.fr

⁴: Department of Economics and Management, University of Brescia
25122 Brescia, Italy
grazia.speranza@unibs.it

Abstract: We address a Multi-Commodity two-echelon Distribution Problem (MC2DP) where three sets of stakeholders are involved: suppliers, distribution centers, and customers. Multiple commodities have to be sent from suppliers to customers, using multiple distribution centers for consolidation purposes. Commodities are collected from the suppliers and delivered to the distribution centers with direct trips, while a fleet of homogeneous vehicles distributes commodities to customers. Commodities are compatible, that is any vehicle can transport any set of commodities as long as its capacity is not exceeded. The goal is to minimize the total transportation cost from suppliers to customers. We present two sequential schemes based on the solution, in a different order, of a collection and a delivery subproblem. In both cases, the solution of the first subproblem determines the quantity of each commodity at each distribution center. The second subproblem takes this information as input. We also propose different strategies to guide the solution of the first subproblem in order to take into account the impact of its solution on the second subproblem. The proposed sequential heuristics are evaluated and compared both on randomly generated instances and on a case study related to a short and local fresh food supply chain. The results show the impact of problem characteristics on solution strategies.

Keywords: multicommodity, routing problem, local fresh food supply chain, sequential solution.

1 Introduction

In this paper we study a complex distribution problem in a two-echelon supply chain where three sets of stakeholders are involved: suppliers, distribution centers and customers. Multiple commodities are collected from the suppliers and delivered to the customers through distribution centers for consolidation purposes. Each supplier has a given available quantity for each commodity and each customer has a demand for each commodity.

We consider a single decision maker who manages all distribution centers and organizes the collection and delivery operations. The commodities are collected from suppliers and delivered to distribution centers through direct trips, and distributed from the distribution centers to customers with a fleet of vehicles performing routes. Direct deliveries from suppliers to customers are not allowed. Commodities are compatible, that is, any vehicle can transport any set of commodities as long as its capacity is not exceeded. Multiple visits to a customer are allowed to reduce transportation costs. However, for the sake of customers convenience, a single commodity has to be delivered at once. The problem is named Multi-Commodity two-echelon Distribution Problem (MC2DP). The objective is to find a collection and delivery plan that minimizes the total transportation cost, satisfying customer demands, and not exceeding the available quantities at the suppliers and the vehicle capacities. The MC2DP is a problem belonging to the operational decision level. The plan covers one day only, i.e., collection and delivery operations performed during a day. The study of this problem is motivated by a case study presented in Section 6 for the collection and delivery of fresh agri-food products (fruits and vegetables) through a short and local supply chain.

In the following, we detail the application on short and local fresh food supply chain to motivate our work. The production and delivery of fresh food products have undergone important changes in Europe since the 1950's, especially through the modernization of the tools and processes in order to meet the customer demand with low production costs. Multinational companies have played a major role as intermediaries between farmers and consumers ([Rucabado-Palomar and Cuéllar-Padilla, 2018](#)). Nowadays, one of the major problems faced by farmers is the shortfall of their incomes: over the last decades they have been encouraged to produce more, while their unit selling price was decreasing. However, in many regions there coexist (1) supplies with medium-sized farms where various products of high quality (freshness, few pesticide) are cultivated and (2) customers with a strong desire for product quality and traceability ([King et al., 2015](#)). Hence, the idea has emerged to locally connect suppliers and customers ([Berti and Mulligan, 2016](#)), by means of a short (and/or) local food supply chain. The main purpose of this kind of supply chain is to capture more end-use value for the farmers.

Short food supply chains are defined as an opportunity for agricultural products to reach the market either through direct sales or through indirect sales with only one intermediary between producers and consumers. Local food supply chains may involve several intermediaries with all the actors located on a limited area (e.g. considering geographical or political restrictions). The maximum distance between actors is usually around 80 km ([Blanquart et al., 2010](#)). In this paper, we consider short and local food supply chains with indirect sales to canteens, restaurants or supermarkets.

Short and local supply chains involve few intermediaries. Hence, farmers have to take charge of a large part of their products marketing and distribution, which is not their core business. It is feasible for direct sales since the volumes are usually low, and consequently farmers can spend time selling their products. For indirect sales (canteen, restaurants or supermarket), volumes are more important and, therefore, the supply chain has to rely on a set of distribution centers in order to organize product flows and to minimize transportation costs, with the aim to be competitive with conventional food supply chains. Farmers supply these distribution centers by performing direct trips since the volumes are large. The distribution centers are then in charge of consolidation and delivery of the products to customers. In a local supply chain context, all the actors (farmers, distribution centers and customers) are located in a restricted area. A single decision maker manages all the distribution

centers, and coordinates the transportation planning for both collection and delivery operations. This decision maker can be an association of farmers, or a local political authority. These distribution centers are considered as the only intermediary in the supply chain.

The Multi-Commodity two-echelon Distribution Problem belongs to the broad class of two-echelon routing problems, which are distribution problems where transportation activities take place in two echelons of a supply chain. There exists a wide literature on this class of problems. We refer to [Cuda et al. \(2015\)](#) for a recent survey on two-echelon routing problems and to [Guastaroba et al. \(2016\)](#) for a more general survey on transportation problems with intermediate facilities. As mentioned in [Cuda et al. \(2015\)](#), there are different classes of two-echelon routing problems. The closest to the MC2DP is the 2-Echelon Vehicle Routing Problem (2E-VRP) for which a wide literature exists, as surveyed in [Cuda et al. \(2015\)](#). We refer to [Guimarães et al. \(2019\)](#), [Zhou et al. \(2018\)](#) and [Yu et al. \(2020\)](#) for recent applications of the problem.

The problem we study here contributes to the literature on two-echelon routing problems by explicitly considering multiple commodities that are required by final customers. To the best of our knowledge, multiple commodities have never been considered in two-echelon distribution systems. The title of [Dellaert et al. \(2021\)](#) could make the reader think the opposite. However, in [Dellaert et al. \(2021\)](#) the authors studied a one-to-one pickup and delivery problem in a two-echelon distribution system. In other words, a commodity corresponds to an origin-destination pair. In our paper, instead, we have a many-to-many setting where each commodity can be picked up from any origin (supplier in our case) offering it and delivered to any destination (customer) requiring it. Multiple commodities have been explicitly modeled in two-echelon location problems (see, for example, [Hinojosa et al. \(2000\)](#) and [Sadjady and Davoudpour \(2012\)](#)). These problems do not involve routing decisions but evaluate the cost of assigning final customers to a specific distribution center.

The distribution centers considered in this work can be viewed as cross-docking stations. Using cross-dock platforms is a common practice for fresh products since it allows the transfer of incoming shipments directly to outgoing vehicles, without the need to store them in between ([Bruzzone et al. \(2009\)](#)). Cross-dock platforms are also often used for the delivery from manufacturers in e-commerce, home delivery and urban logistics ([Qiu et al., 2021](#)). The direct consequence of this practice is to have shorter delivery times that help to preserve the freshness of the foods. It is also well-known that cross-docking or consolidation platforms increase the overall performance of a delivery system, independently of the products that go through the platform. They allow cost reduction of the overall logistics service ([Ladier and Alpan, 2016](#), [Van Belle et al., 2012](#)).

Contributions on the management of a cross-docking platform, motivated by the fresh food delivery, exist in the literature. These studies usually deal with the management and coordination of inbound and outbound vehicles and the consolidation of products in outbound vehicles. [Agustina et al. \(2014\)](#) consider the problem of building the delivery plan to serve customers while minimizing inventory and routing costs plus earliness and tardiness delivery penalties. The authors propose a mixed integer linear program that is solved with Cplex. [Rahbari et al. \(2019\)](#) propose a bi-objective model that considers cost minimization as well as product freshness maximization. In both papers, customers' demands do not explicitly consider multiple commodities.

The presence of multiple commodities makes the MC2DP substantially different with respect to the standard 2E-VRP. In fact, the demand matching decision, i.e., the decision related to the assignment of suppliers and customers to distribution centers, is constrained not only by the matching of quantities, but also involves the matching of commodities. In addition, in the MC2DP, customers may be visited more than once, according to the number of commodities they request. An additional consideration is due to synchronization constraints that might be present in two-echelon distribution systems. The two echelons may be linked by different kinds of synchronization constraints (see [Drexler \(2012\)](#)). The most common one considered in the 2E-VRP is the *load synchronization* that establishes that the quantity delivered to each distribution center in the first echelon has to be sufficient to satisfy the

demand of the customers assigned to the distribution center in the second echelon. Another common synchronization constraint is the *operation synchronization*, implying precedence constraints between the collection and the delivery operations. In the MC2DP, only load synchronization is considered, specified for each commodity. Operation synchronization is not considered. In other words, we assume that all collection operations are performed before the delivery operations (for example, collection in the morning and delivery in the afternoon).

The explicit consideration of different commodities is essential in the agri-food supply chains since availability at the producer depends on the production of farmers and requirements made by customers concern specific commodities.

There is a large literature on planning problems in agri-food supply chains. The main focus is, in general, evaluating the level of degradation and the quality of products in the planning models (Yu and Xiao, 2021). When focusing on short and local food supply chain, the literature is very scarce. As pointed out by Flores and Villalobos (2018), there is a lack of supply chain planning tools for local fresh food supply chains. Flores and Villalobos (2018) develop an agricultural planning framework that determines an optimal production planning of vegetables for local supply chains. Bosona and Gebresenbet (2011) address a case study in Sweden. They first study the location of distribution centers using the centre-of-gravity technique. Then, farmers are assigned to one distribution center, and can bring their products to the distribution center, or the latter can perform collection by grouping farmers into routes. Routes are then optimized using a dedicated software (Route LogiX). Ogier et al. (2013) propose a mixed integer linear programming model for service network design of a short and local supply chain.

Even though the application presented in this paper deals with the agri-food supply chain, the methods we propose can be applied to MC2DPs that consider direct delivery trips in the first echelon and allow splitting the delivery of different commodities in the second level.

In this paper, we present a solution approach to the MC2DP. Due to the complexity of the problem, we propose a sequential approach that first decomposes the MC2DP in two subproblems, associated with the collection and delivery phases, respectively, and then sequentially solves them. Two sequential schemes are presented, depending on which of the two subproblems is solved first. In both cases, the solution of the first subproblem determines the quantity of each commodity at each distribution center. The second subproblem takes this information as input. We also propose different strategies to guide the solution of the first subproblem in order to take into account the impact of its solution on the second subproblem. It is worth noting that the subproblem associated with the delivery phase is itself a new problem. It is a Vehicle Routing Problem (VRP) with multiple commodities and multiple depots, with a maximum available quantity of each commodity at the depots. The VRP with multiple depots is a well-studied problem in the literature (Montoya-Torres et al., 2015). In this paper, the delivery subproblem is an extension of the VRP with multiple depots where several commodities are available at each depot with given quantities. A further contribution of this paper is a solution method for the delivery subproblem, based on the Adaptive Large Neighborhood Search (ALNS) algorithm proposed for the one depot case by Gu et al. (2019). The two proposed sequential heuristic schemes and the different strategies are evaluated and compared both on randomly generated instances, with different characteristics (supplier locations, customer locations, maximum supply quantities), and on a case study for the collection and delivery of fresh food products (fruits and vegetables) through a short and local supply chain using a set of distribution centers located in the French department of Isère. The size of the instances for this case study is large, and two kinds of customers are considered: school canteens and supermarkets. The computational results show the impact of the instance on the solution approaches and strategies.

The remainder of this paper is organized as follows. A definition of the MC2DP is given in Section 2, as well as the problem formulation. A decomposition of the problem into a collection and a delivery subproblems is presented in Section 3. Then, the sequential approach is described in Section 4. Section

5 provides the computational results. The case study on short and local fresh food supply chains in the French department of Isère is discussed in Section 6. Finally, conclusions and prospects are presented in Section 7.

2 Problem definition and formulation

The Multi-Commodity two-echelon Distribution Problem (MC2DP) is defined on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, in which \mathcal{V} is the set of vertices and \mathcal{A} is the set of arcs. More precisely, \mathcal{V} is defined as $\mathcal{V}_S \cup \mathcal{V}_D \cup \mathcal{V}_C$ where $\mathcal{V}_S = \{1, \dots, N_S\}$ represents the set of N_S suppliers, $\mathcal{V}_D = \{N_S + 1, \dots, N_S + N_D\}$ is the set of N_D distribution centers and $\mathcal{V}_C = \{N_S + N_D + 1, \dots, N_S + N_D + N_C\}$ represents the set of N_C customers. We only consider direct trips from suppliers to distribution centers. Direct deliveries from suppliers to customers are not allowed. Moreover, transfers of commodities between distribution centers are not considered. Thus, $\mathcal{A} = \{(i, j), (j, i) | i \in \mathcal{V}_S, j \in \mathcal{V}_D\} \cup \{(i, j), (j, i) | i \in \mathcal{V}_D, j \in \mathcal{V}_C\} \cup \{(i, j) | i, j \in \mathcal{V}_C\}$ is the set of arcs.

Suppliers provide a set of commodities \mathcal{M} , which are transported to the distribution centers by an unlimited fleet of homogeneous vehicles of capacity Q_S . Each supplier $s \in \mathcal{V}_S$ is associated with a maximum available quantity $O_{sm} \geq 0$ of commodity $m \in \mathcal{M}$. Each distribution center has its own unlimited fleet of homogeneous vehicles of capacity Q_D that are used to deliver the commodities to the customers. Customer $i \in \mathcal{V}_C$ requires a quantity $D_{im} \geq 0$ of each commodity $m \in \mathcal{M}$. For each customer $i \in \mathcal{V}_C$, $\mathcal{M}_i = \{m \in \mathcal{M} | D_{im} > 0\}$ represents the set of commodities required by customer i . Commodities are compatible, i.e., they can be transported on the same vehicle. The demand of a customer can be split, that is the customer can be served by several vehicles. However, for the sake of customer convenience, the split policy is constrained: each commodity has to be delivered by one vehicle only. Without loss of generality, it is assumed that $\sum_{m \in \mathcal{M}} D_{im} \leq Q_D$ for all customers $i \in \mathcal{V}_C$. Note that, in case there is a customer i such that $\sum_{m \in \mathcal{M}} D_{im} > Q_D$, then we split the customer and create as many copies as the number of commodities requested by i .

A cost c_{ij} is associated with each arc $(i, j) \in \mathcal{A}$ and represents the non-negative cost of traversing arc (i, j) . In the MC2DP, the decision maker is the logistic provider who manages all the distribution centers. He decides how to collect commodities from the suppliers and how to distribute the commodities from the distribution centers to the customers. The objective is to find a collection and delivery plan that minimizes the total transportation cost, satisfying customer demands, not exceeding the available quantities at the suppliers and the vehicle capacities.

The MC2DP is NP-hard as it reduces to the Traveling Salesman Problem (TSP) in the case where $|\mathcal{M}| = |\mathcal{V}_D| = |\mathcal{V}_S| = 1$ and vehicles have unlimited capacity.

In the following, we will call *collection* the transportation of commodities from suppliers to distribution centers and *delivery* the distribution of commodities from distribution centers to customers. Note that, in the following, we will use the word *truck* to indicate a vehicle used for collection operations, to differentiate it from a vehicle used for delivery operations. The collection and delivery phases of the MC2DP are connected through the quantities of commodities at the distribution centers. We denote by U_{dm} the unknown quantity of commodity $m \in \mathcal{M}$ at distribution center d .

An example of an instance of the MC2DP is depicted in Figure 1. There are two commodities, the truck capacity is $Q_S = 8$ and the vehicle capacity $Q_D = 10$. A feasible solution of this instance is shown in Figure 2. In the solution, five trucks are used to supply the distribution centers, each truck performing a direct trip. The numbers on the arcs representing direct trips report the quantities transported for each commodity, where the number corresponding to the commodity is reported in parentheses. Note that two trucks are used from supplier 1 to distribution center 4. Quantities U_{dm} are as follows: $U_{41} = 4$, $U_{42} = 14$, $U_{51} = 5$ and $U_{52} = 3$. For the distribution to customers, two vehicles leave distribution center 4, while one vehicle leaves distribution center 5. Note that customer 9 is visited twice: one vehicle delivers commodity 1 while another vehicle delivers commodity 2.

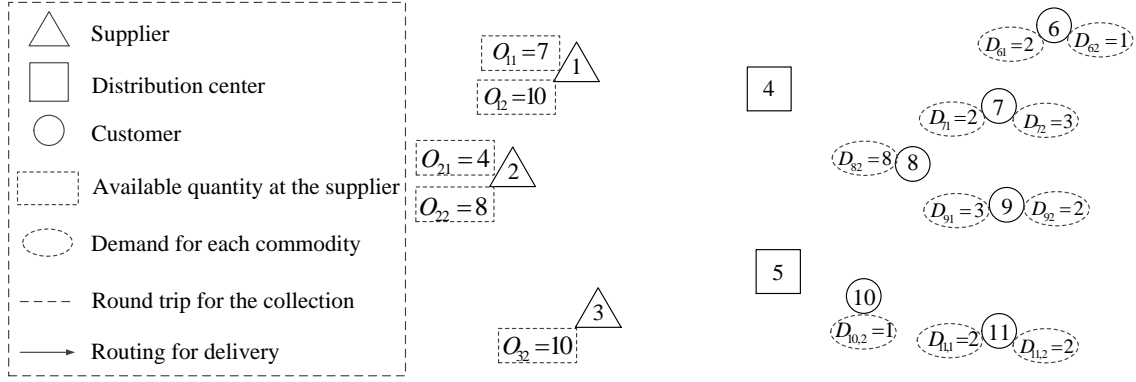


Figure 1: An instance of the MC2DP.

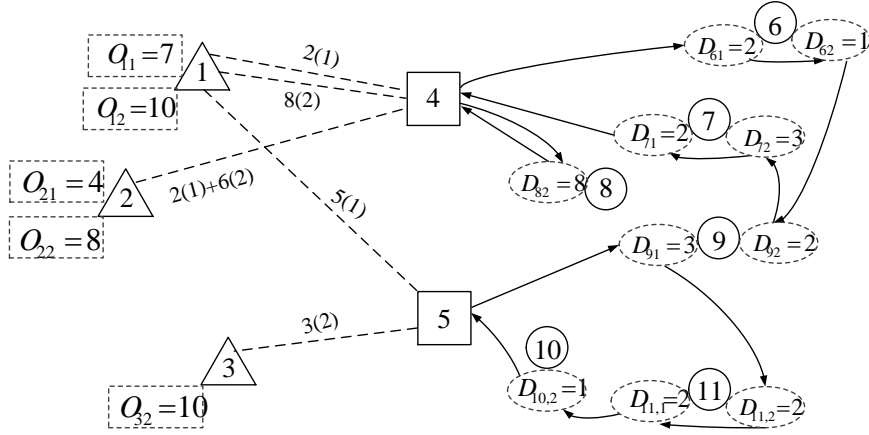


Figure 2: A feasible solution of the MC2DP instance.

2.1 Problem formulation

We now propose a mathematical formulation of the problem. Let us indicate with \mathcal{F}_d the fleet associated with the delivery from distribution center $d \in \mathcal{V}_D$. Since this fleet in the MC2DP is supposed to be unlimited, we set $|\mathcal{F}_d| = N_C, \forall d \in \mathcal{V}_D$. This is clearly an upper bound on the number of vehicles needed at each distribution center, since $\sum_{m \in \mathcal{M}} D_{im} \leq Q_D$ for all $i \in \mathcal{V}_C$.

Let us number the vehicles such that $\mathcal{F}_1 = \{1, \dots, N_C\}, \mathcal{F}_2 = \{N_C + 1, \dots, 2N_C\}, \dots, \mathcal{F}_{N_D} = \{N_C(N_D - 1) + 1, \dots, N_D N_C\}$. Let $\mathcal{F} = \cup_{d=1}^{N_D} \mathcal{F}_d$. Note that labelling the vehicles by increasing index allows us to drop the distribution center index in the variable definition.

Let us indicate with $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$, where $\mathcal{A}_1 = \{(i, j) | i \in \mathcal{V}_S, j \in \mathcal{V}_D\}$ and $\mathcal{A}_2 = \{(i, j), (j, i) | i \in \mathcal{V}_D, j \in \mathcal{V}_C\} \cup \{(i, j) | i, j \in \mathcal{V}_C\}$. Let us also define $D_{im}^- = \min_{m \in \mathcal{M}_i} \{D_{im}\}$ and $d(f)$ as the distribution center $d \in \mathcal{V}_D$ associated with vehicle $f \in \mathcal{F}$.

The following decision variables are introduced.

- x_{sd}^1 : non-negative integer variable equal to the number of trucks traversing arc $(s, d) \in \mathcal{A}_1, s \in \mathcal{V}_S, d \in \mathcal{V}_D$.
- x_{ijf}^2 : binary variable equal to 1 if arc $(i, j) \in \mathcal{A}_2$ is traversed by vehicle $f \in \mathcal{F}$.
- z_{if} : binary variable equal to 1 if customer or distribution center $i \in \mathcal{V}_C \cup \mathcal{V}_D$ is visited by vehicle $f \in \mathcal{F}$.
- u_{if} : non-negative real variable equal to the accumulated demand to be delivered by vehicle $f \in \mathcal{F}$ when arriving at customer $i \in \mathcal{V}_C$.

- y_{if}^m : binary variable equal to 1 if commodity $m \in \mathcal{M}$ of customer $i \in \mathcal{V}_C$ is delivered by vehicle $f \in \mathcal{F}$.
- q_{sd}^m : non-negative variable equal to the quantity of commodity $m \in \mathcal{M}$ picked up at supplier $s \in \mathcal{V}_S$ and delivered to distribution center $d \in \mathcal{V}_D$.

In addition, we define as $\bar{c}_{sd} = c_{sd} + c_{ds}$ the direct trip cost between supplier $s \in \mathcal{V}_S$ and distribution center $d \in \mathcal{V}_D$.

The compact formulation for the MC2DP, denoted as \mathcal{F}_{MC2DP} , is as follows:

$$(\mathcal{F}_{MC2DP}) \min \sum_{(s,d) \in \mathcal{A}_1} \bar{c}_{sd} x_{sd}^1 + \sum_{(i,j) \in \mathcal{A}_2} \sum_{f \in \mathcal{F}} c_{ij} x_{ijf}^2 \quad (1)$$

$$\text{s.t.} \sum_{f \in \mathcal{F}_d} \sum_{i \in \mathcal{V}_C} y_{if}^m D_{im} \leq \sum_{s \in \mathcal{V}_S} q_{sd}^m, \quad \forall d \in \mathcal{V}_D, \forall m \in \mathcal{M}, \quad (2)$$

$$\sum_{m \in \mathcal{M}} q_{sd}^m \leq Q_S x_{sd}^1, \quad \forall s \in \mathcal{V}_S, \forall d \in \mathcal{V}_D, \quad (3)$$

$$\sum_{d \in \mathcal{V}_D} q_{sd}^m \leq O_{sm}, \quad \forall s \in \mathcal{V}_S, \forall m \in \mathcal{M}, \quad (4)$$

$$\sum_{f \in \mathcal{F}} y_{if}^m = 1, \quad \forall i \in \mathcal{V}_C, \forall m \in \mathcal{M}_i, \quad (5)$$

$$\sum_{j|(i,j) \in \mathcal{A}_2} x_{ijf}^2 = \sum_{j|(j,i) \in \mathcal{A}_2} x_{jif}^2 = z_{if}, \quad \forall i \in \mathcal{V}_C \cup \mathcal{V}_D, \forall f \in \mathcal{F}, \quad (6)$$

$$\sum_{f \in \mathcal{F} \setminus \mathcal{F}_d} z_{df} = 0, \quad \forall d \in \mathcal{V}_D, \quad (7)$$

$$y_{if}^m \leq z_{if}, \quad \forall i \in \mathcal{V}_C, \forall m \in \mathcal{M}_i, \forall f \in \mathcal{F}, \quad (8)$$

$$\sum_{i \in \mathcal{V}_C} \sum_{m \in \mathcal{M}_i} D_{im} y_{if}^m \leq Q_D z_{d(f)f}, \quad \forall f \in \mathcal{F}, \quad (9)$$

$$u_{if} - u_{jf} + Q_D x_{ijf}^2 \leq Q_D - D_{jm}^-, \quad \forall (i,j) \in \mathcal{A}_2, \forall f \in \mathcal{F}, \quad (10)$$

$$x_{sd}^1 \in \mathbb{N}, \quad \forall (s,d) \in \mathcal{A}_1, \quad (11)$$

$$x_{ijf}^2 \in \{0, 1\}, \quad \forall (i,j) \in \mathcal{A}_2, \forall f \in \mathcal{F}, \quad (12)$$

$$z_{if} \in \{0, 1\}, \quad \forall i \in \mathcal{V}_C \cup \mathcal{V}_D, \forall f \in \mathcal{F}, \quad (13)$$

$$y_{if}^m \in \{0, 1\}, \quad \forall i \in \mathcal{V}_C, f \in \mathcal{F}, \forall m \in \mathcal{M}, \quad (14)$$

$$u_{if} \geq 0, \quad \forall i \in \mathcal{V}_C, \forall f \in \mathcal{F}, \quad (15)$$

$$q_{sd}^m \geq 0, \quad \forall (s,d) \in \mathcal{A}_1, \forall m \in \mathcal{M}. \quad (16)$$

The objective function (1) minimizes the total transportation costs. Constraints (2) impose that the quantity of each commodity delivered to a distribution center is sufficient to carry out delivery operations from there. Constraints (3) impose the use of a sufficient number of trucks for collection operations. Constraints (4) fix the maximum availability of each commodity at each supplier. Constraints (5) state that each required commodity is distributed exactly once to each customer. Constraints (6) are flow conservation constraints. Constraints (7) manage the assignment of vehicles to distribution centers. Constraints (8) impose to use a vehicle if this delivers at least one commodity of a customer. Constraints (9) are vehicle capacity constraints. Constraints (10) eliminate subtours that do not contain a distribution center. Constraints (11)–(16) define the variables.

3 Problem decomposition

The MC2DP integrates two subproblems, one considering the collection of commodities, that is the transportation of the commodities from the suppliers to the distribution centers, the other considering the delivery of commodities from the distribution centers to the customers. In this section, we present a decomposition of the MC2DP and define the two subproblems: the SPC (SubProblem Collection) and the SPD (SubProblem Delivery). The connection of these two problems is made by quantities U_{dm} that represent the available quantity of commodity m at distribution center d . For the sake of clarity, when necessary, we will denote by U_{dm}^D the quantity of commodity m that is delivered from the distribution center d to customers, and by U_{dm}^C the quantity of commodity m that is collected at the suppliers and delivered to the distribution center d . Figure 3 illustrates the solutions of the SPC and the SPD for the instance in Figure 2, with the associated U_{dm} variables that link the two subproblems.

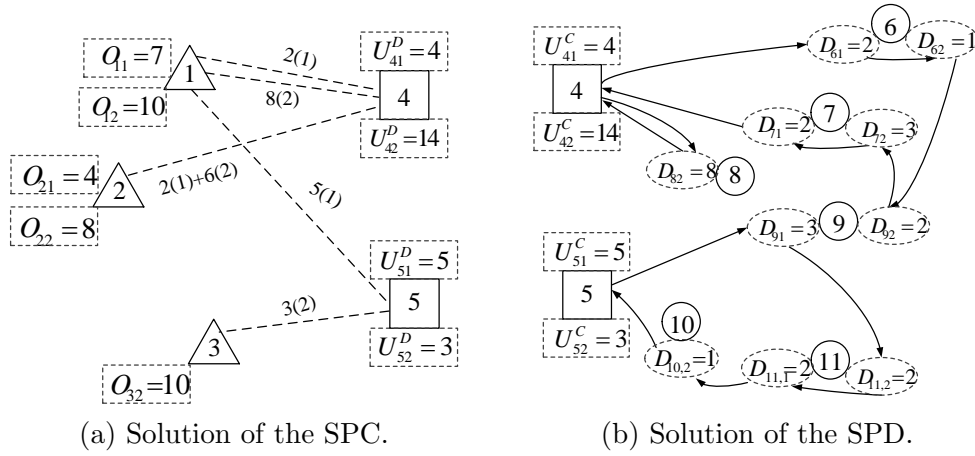


Figure 3: Solutions of the two subproblems based on the solution of the MC2DP in Figure 2.

We now provide a formal definition of the SPC and the SPD. Note that each subproblem is concerned with the optimization of the related operations, i.e., SPC minimizes the cost of collection operations only while SPD minimizes the cost of delivery operations only.

3.1 The collection subproblem (SPC)

The SPC is defined on a graph $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{A}_1)$, where $\mathcal{V}_1 = \mathcal{V}_S \cup \mathcal{V}_D$ and $\mathcal{A}_1 = \{(i, j), (j, i) | i \in \mathcal{V}_S, j \in \mathcal{V}_D\}$. The SPC consists in determining a set of direct trips for trucks between suppliers and distribution centers with the associated quantities for each commodity. The objective is to minimize the transportation cost, defined as the total cost for the direct trips, that is independent of the quantity transported on each truck. For each commodity, the quantity transported to distribution centers has to be sufficient to satisfy customer demands in the SPD. Moreover, the solution of the SPC must satisfy the following constraints:

- (1) the total quantity of commodities transported by each truck does not exceed the truck capacity Q_S ;
- (2) for each supplier s , the quantity of each commodity m that is transported to distribution centers must be at most equal to the available quantity O_{sm} ;
- (3) the quantity of each commodity m transported to each distribution center d is greater than the required quantity U_{dm}^D .

When the SPC is solved first, the values U_{dm}^D are not known and have to be set to a valid lower bound. Next, the values U_{dm}^C are computed from the solution of the SPC and used as input to the SPD.

The SPC is related to the Multi-commodity Capacitated fixed-charge Network Design problem (MCND) (Magnanti and Wong, 1984). The MCND problem is a discrete optimization problem in which a set of commodities has to be routed through a directed network. Each commodity has a demand to be transported from an origin to a destination. Each arc has a limited capacity, a unit cost flow, and a fixed cost if the arc is used (the flow is positive). The SPC differs from the MCND problem because of the cost structure. In the SPC, the cost of an arc depends on the number of trucks used: it is a step-wise cost function defined by a unitary cost associated with each truck used. Thus, steps are defined as multiples of the truck capacity Q_S .

3.2 The delivery subproblem (SPD)

The SPD is defined on a graph $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{A}_2)$ where $\mathcal{V}_2 = \mathcal{V}_D \cup \mathcal{V}_C$ and $\mathcal{A}_2 = \{(i, j), (j, i) | i \in \mathcal{V}_D, j \in \mathcal{V}_C\} \cup \{(i, j) | i, j \in \mathcal{V}_C\}$. In the following, we will use the word *vehicle* to indicate a vehicle used in the SPD.

The SPD consists in assigning commodities to vehicles and in determining a set of vehicle routes to meet all customer demands. The solution must satisfy the following constraints:

- (1) the total quantity of commodities delivered by each vehicle does not exceed the vehicle capacity Q_D ;
- (2) each commodity requested by each customer is delivered by a single vehicle;
- (3) the demand of all customers is satisfied;
- (4) the quantity of each commodity m distributed from each distribution center d does not exceed the available quantity U_{dm}^C ;
- (5) each vehicle starts and ends its route at the same distribution center.

When the SPD is solved first, the values U_{dm}^C are not known and have to be set to a valid upper bound. Next, the values U_{dm}^D are computed from the solution of the SPD and used as input to the SPC.

The SPD is the multi-depot case of the Commodity constrained Split Delivery Vehicle Routing Problem (C-SDVRP) (Archetti et al., 2014). The C-SDVRP is a problem where customers require multiple commodities. Each customer can be served by different vehicles, but each commodity has to be delivered at once by a single vehicle. The C-SDVRP considers only one distribution center with sufficient quantity for each commodity to satisfy all the customer demands.

4 A sequential approach

In this section we describe the sequential solution approach for the MC2DP. We first describe a solution method for the SPC based on the solution of a mathematical programming model. Then, we propose a solution method for the SPD. It is an Adaptive Large Neighborhood Search (ALNS) algorithm that extends the approach described in Gu et al. (2019). As the solution of the first subproblem solved, whatever it is, has an impact on the solution of the second subproblem, we describe, in Section 4.3, different strategies for taking into account this impact when solving the first subproblem.

We will make use of the concept of *customer-commodity*. A *customer-commodity* represents the demand of a customer for a single commodity. A customer may be seen as a union of *customer-commodities*.

4.1 Solution of the SPC

The SPC is modelled as a Mixed Integer linear Program (MIP). The formulation, that will be solved to optimality, is based on the following decision variables:

- x_{sd} : non-negative integer variable equal to the number of trucks traversing arc $(s, d) \in \mathcal{A}_1, s \in \mathcal{V}_S, d \in \mathcal{V}_D$;
- q_{sd}^m : non-negative variable equal to the quantity of commodity $m \in \mathcal{M}$ picked up at supplier $s \in \mathcal{V}_S$ and delivered to distribution center $d \in \mathcal{V}_D$;
- y_{di}^m : binary variable equal to 1 if the demand of commodity $m \in \mathcal{M}$ of customer $i \in \mathcal{V}_C$ is assigned to distribution center $d \in \mathcal{V}_D$, and 0 otherwise.

Note that the variables x_{sd} and q_{sd}^m have the same meaning as variables x_{sd}^1 and q_{sd}^m , respectively, in the formulation \mathcal{F}_{MC2DP} . Variables y_{di}^m are introduced for the formulation of the SPC.

The mathematical formulation that follows determines the quantities to transport from suppliers to distribution centers in such a way that all commodities requested by customers are shipped, and the availabilities at the suppliers and the truck capacity are not violated:

$$\min \sum_{(s,d) \in \mathcal{A}_1} \bar{c}_{sd} x_{sd} \quad (17)$$

$$\text{s.t. } \sum_{d \in \mathcal{V}_D} y_{di}^m = 1, \quad \forall i \in \mathcal{V}_C, m \in \mathcal{M}_i, \quad (18)$$

$$\sum_{i \in \mathcal{V}_C} D_{im} y_{di}^m \leq \sum_{s \in \mathcal{V}_S} q_{sd}^m, \quad \forall d \in \mathcal{V}_D, m \in \mathcal{M}, \quad (19)$$

$$\sum_{s \in \mathcal{V}_S} q_{sd}^m \geq U_{dm}^D, \quad \forall d \in \mathcal{V}_D, m \in \mathcal{M}, \quad (20)$$

$$\sum_{m \in \mathcal{M}} q_{sd}^m \leq Q_S x_{sd}, \quad \forall s \in \mathcal{V}_S, d \in \mathcal{V}_D, \quad (21)$$

$$\sum_{d \in \mathcal{V}_D} q_{sd}^m \leq O_{sm}, \quad \forall s \in \mathcal{V}_S, m \in \mathcal{M}, \quad (22)$$

$$x_{sd} \in \mathbb{N}, \quad \forall s \in \mathcal{V}_S, d \in \mathcal{V}_D, \quad (23)$$

$$y_{di}^m \in \{0, 1\}, \quad \forall i \in \mathcal{V}_C, d \in \mathcal{V}_D, m \in \mathcal{M}, \quad (24)$$

$$q_{sd}^m \geq 0, \quad \forall s \in \mathcal{V}_S, d \in \mathcal{V}_D, m \in \mathcal{M}. \quad (25)$$

The objective function (17) minimizes the transportation cost, that is the cost of the direct trips of the trucks used to supply the distribution centers. Constraints (18) impose that, for each customer, the demand of each commodity is assigned to one distribution center. Constraints (19) impose that the quantity of each commodity delivered to each distribution center satisfies the demand of the customers assigned to the same distribution center. Note that constraints (18) and (19) ensure that any solution of the SPC is feasible for the SPD. Constraints (20) ensure that the quantity transported to each distribution center is greater than the lower bound U_{dm}^D . Constraints (21) impose that the total volume of all commodities transported from a supplier to a distribution center cannot exceed the capacity of the trucks used on that arc. Constraints (22) impose not to exceed the available quantities of commodities at the suppliers. Constraints (23)–(25) define the variables.

The above formulation may have several equivalent optimal solutions. To break ties among solutions with the same cost but delivering different quantities, we consider a small value ε and add to the objective function (17) the term:

$$\varepsilon \sum_{d \in \mathcal{V}_D} \sum_{i \in \mathcal{V}_C} \sum_{m \in \mathcal{M}_i} c_{di} y_{di}^m, \quad (26)$$

where c_{di} is the travelling cost between distribution center d and customer i . With this term, solutions with different assignments of *customer-commodities* to distribution centers are evaluated differently. This term favours solutions that minimize the total distance between *customer-commodities* and the distribution center to which they are assigned.

We call \mathcal{F}_{SPC} the formulation (18)-(25) with objective function:

$$\min \sum_{(s,d) \in \mathcal{A}_1} \bar{c}_{sd} x_{sd} + \varepsilon \sum_{d \in \mathcal{V}_D} \sum_{i \in \mathcal{V}_C} \sum_{m \in \mathcal{M}_i} c_{di} y_{di}^m. \quad (27)$$

An optimal solution to the formulation \mathcal{F}_{SPC} gives the available quantities U_{dm}^C of each commodity m at distribution center d as:

$$U_{dm}^C = \sum_{s \in \mathcal{V}_S} q_{sd}^m.$$

We will see in Section 4.3, how such quantities are used in the sequential approach for the solution of the MC2DP where the SPC is solved first. When the SPD is solved first, the assignment of *customer-commodities* to distribution centers is known, that is the values of the y variables are known. In this case, constraints (18) and (19) are removed from \mathcal{F}_{SPC} , and only constraints (20) are used to link the two subproblems. It is noteworthy that when the SPC is solved first, the values U_{dm}^D are not set according to the solution of the SPD but according to the strategies defined in Section 4.3.2. Then, constraints (18) and (19) are necessary to ensure that the solution of the SPC is feasible for the SPD.

4.2 Solution of the SPD

The SPD deals with the delivery of commodities from distribution centers to customers. As already mentioned, when the SPD is solved after the SPC, the values U_{dm}^C are known. Instead, when the SPD is solved first, the values U_{dm}^C have to be set. Different strategies to determine these values are proposed in Section 4.3.2. In the remaining of this section we assume that these quantities U_{dm}^C are given, either from the solution of the SPC or from one of the strategies described in Section 4.3.1, when the SPD is solved first.

In the following, we describe the solution approach we propose for the SPD, which is an Adaptive Large Neighborhood Search (ALNS). Section 4.2.1 describes how we build an initial solution while the ALNS is described in Section 4.2.2.

4.2.1 Initial solution

To solve the SPD, we start by assigning *customer-commodities* to distribution centers. To this end, we solve a Generalized Assignment Problem (GAP) (Ross and Soland (1975), Cattrysse and Van Wassenhove (1992)) which aims at finding the minimum cost assignment of *customer-commodities* to distribution centers so that each *customer-commodity* is assigned to exactly one distribution center. The assignment is subject to the availability of each commodity at the distribution centers. The assignment cost of commodity m for customer i to distribution center d is c_{id} that represents the traveling cost from customer i to distribution center d .

We now present the formulation of the GAP. Let x_{id}^m be the binary variables equal to 1 if and only if commodity $m \in \mathcal{M}$ required by customer $i \in \mathcal{V}_C$ is served from distribution center $d \in \mathcal{V}_D$, and 0 otherwise. The GAP is formulated as follows:

$$\min \sum_{i \in \mathcal{V}_C} \sum_{m \in \mathcal{M}_i} \sum_{d \in \mathcal{V}_D} c_{id} x_{id}^m \quad (28)$$

$$\text{s.t. } \sum_{i \in \mathcal{V}_C} D_{im} x_{id}^m \leq U_{dm}^C, \quad \forall m \in \mathcal{M}, d \in \mathcal{V}_D, \quad (29)$$

$$\sum_{d \in \mathcal{V}_D} x_{id}^m = 1, \quad \forall i \in \mathcal{V}_C, m \in \mathcal{M}_i, \quad (30)$$

$$x_{id}^m \in \{0, 1\}, \quad \forall i \in \mathcal{V}_C, m \in \mathcal{M}, d \in \mathcal{V}_D. \quad (31)$$

The objective function (28) minimizes the total assignment cost of *customer-commodities* to distribution centers. Constraints (29) impose not to exceed the quantity of commodity available at each distribution center. Constraints (30) ensure that each *customer-commodity* is assigned to exactly one distribution center. Constraints (31) define decision variables.

Given the assignment of *customer-commodities* to distribution centers provided by the solution of the GAP, the set of initial routes is obtained by applying the split algorithm to each distribution center. The split algorithm (Beasley (1983), Prins (2004)) starts from a giant-tour that visits all the customers associated with a distribution center and decomposes the visiting sequence into a set of feasible routes. The solution obtained is the initial solution for the ALNS.

4.2.2 The Adaptive Large Neighborhood Search

The basic idea of an ALNS is to improve the current solution by destroying it and building a new one. The ALNS algorithm we present adopts the scheme proposed in Gu et al. (2019) for the commodity constrained split delivery VRP with a single depot (distribution center). The presence, in the problem we solve, of multiple depots and different quantities of each commodity in each of them required substantial changes and the design of ad-hoc procedures. The basic scheme of the algorithm is given in Algorithm 1, where $f(s)$ is the objective value of solution s .

Algorithm 1 The ALNS algorithm.

```

1:  $s \leftarrow$  initial solution
2:  $s_{best} \leftarrow$  initial solution
3: repeat
4:    $s \leftarrow$  Destroy and Repair( $s$ )
5:    $s \leftarrow$  Local Search( $s$ )
6:   if  $f(s) < f(s_{best})$  then
7:      $s \leftarrow$  Intensification( $s$ )
8:      $s_{best} \leftarrow s$ 
9:   end if
10: until stopping criterion is met

```

In the following we present the main features of the algorithm. We allow and penalize violations of the vehicle capacity in the cost function when *customer-commodities* are inserted into existing routes. A maximum capacity violation on each route is imposed. Hence, it is possible that a *customer-commodity* cannot be inserted in any route of the current solution. In this case, we select one distribution center from which serving this *customer-commodity* is feasible with respect to the availability, and we create an additional route. If several distribution centers may be chosen, the one that minimizes the delivery cost is selected. We also allow violations of the limited quantities U_{dm}^C of

commodities available at each distribution center. Both the violation of vehicle capacity and quantities U_{dm}^C are penalized in the objective function, proportionally to the violation. The penalty rates are denoted as λ and γ , respectively. Hence, if a solution violates the vehicle capacity (available quantities U_{dm}^C) by an amount v , then a term λv (γv) is added to the cost function. Initially, λ (γ) is set to a minimum value λ_{min} (γ_{min}) that is equal to the cost of the initial solution. The penalty rate λ (γ) is then dynamically modified during the search as follows. We keep track of the number of consecutive feasible and infeasible solutions visited during the ALNS algorithm. If E_{inf} infeasible solutions are obtained consecutively, the value of λ (γ) is increased to 2λ (2γ). Similarly, if E_{feas} feasible solutions are generated consecutively, the value of λ (γ) is decreased to $\max\{\lambda_{min}; \lambda/2\}$ ($\max\{\gamma_{min}; \gamma/2\}$).

Destroy and Repair. This procedure aims at diversifying the search. It relies on a set of removal and insertion operators which iteratively destroy and repair solutions. The removal and insertion operators are selected using a roulette wheel mechanism. The probability of selecting an operator is dynamically influenced by its performance in past iterations (Ropke and Pisinger, 2006). It uses two removal operators (Shaw removal and random removal of customers), and three insertion heuristics of *customer-commodity* based on greedy, regret-2 and regret-3 insertion paradigms.

Local search. The Local Search procedure (LS) considers classical operators (Laporte (2009), Vidal et al. (2013), Laporte et al. (2014)): insertion, swap and 2-opt for customers, and insertion and swap for *customer-commodities*. Insertion consists in relocating a customer or a *customer-commodity* from its current position in a different position of the same or another route. Swap consists in exchanging the position of two customers or customer-commodities. 2-opt replaces arcs (u, x) and (v, y) in a route by arcs (u, v) and (x, y) , where u and v represent customers and x and y their successors. We do not consider the 2-opt operator based on customer-commodities. In fact this operator works on elements of the same route and it is never beneficial to split apart customer-commodities associated with the same customer. Note that when a move is applied on a customer (and not on a *customer-commodity*), it means that all commodities delivered to the customer in the route are involved in the move. These operators consider moves in the same route, moves between different routes assigned to the same distribution center and moves between routes assigned to different distribution centers.

Intensification. When a new best solution is found, we intensify the search by applying a Mathematical Programming Operator (MPO). The main goal is to define a new assignment of the visits to a customer i by solving a capacitated facility location problem. The MPO has been introduced in Gu et al. (2019) and we modified it for the solution of the SPD. For the sake of clarity, we provide in the following the formulation of the MPO. We introduce the following notation:

- s_i : solution obtained from the current solution by removing all the visits to customer i ;
- \mathcal{R}_i : set of routes in s_i ;
- C_r^i : cost for inserting customer i in route $r \in \mathcal{R}_i$ (cheapest insertion);
- Q_r^i : remaining capacity in route $r \in \mathcal{R}_i$.

The decision variables are the following:

- x_{mr}^i : binary variable equal to 1 if the delivery of commodity $m \in \mathcal{M}_i$ of customer $i \in \mathcal{V}_C$ is assigned to route $r \in \mathcal{R}_i$;
- x_r^i : binary variable equal to 1 if customer $i \in \mathcal{V}_C$ is inserted in route $r \in \mathcal{R}_i$.

The formulation of the MPO is the following:

$$(IP_{MPO}) \min \sum_{r \in \mathcal{R}_i} C_r^i x_r^i \quad (32)$$

$$\text{s.t. } \sum_{r \in \mathcal{R}_i} x_{mr}^i = 1, \quad \forall m \in \mathcal{M}_i, \quad (33)$$

$$\sum_{m \in \mathcal{M}_i} D_{im} x_{mr}^i \leq Q_r^i x_r^i, \quad \forall r \in \mathcal{R}_i, \quad (34)$$

$$x_{mr}^i \in \{0, 1\}, \quad \forall m \in \mathcal{M}_i, r \in \mathcal{R}_i, \quad (35)$$

$$x_r^i \in \{0, 1\}, \quad \forall r \in \mathcal{R}_i. \quad (36)$$

The objective function (32) aims at minimizing the total insertion cost. Constraints (33) require that each commodity is assigned to one route. Constraints (34) impose that the total quantity of commodities assigned to a selected vehicle does not exceed its capacity. Constraints (35)–(36) define the decision variables. (IP_{MPO}) is solved for each $i \in \mathcal{V}_C$ and only the reassignment of visits of customer i associated with the largest cost reduction is implemented.

Formulation (32)–(36) corresponds to the one proposed in Gu et al. (2019). In order to solve the SPD, we add another constraint ensuring that the distribution centers are not overloaded. Specifically, let Q_{dm}^i represent the remaining quantity of commodity m at depot $d \in \mathcal{V}_D$ in s_i . Then, the following constraints are added:

$$D_{im} x_{mr}^i \leq Q_{dm}^i, \quad \forall m \in \mathcal{M}_i, r \in \mathcal{R}_i, \quad (37)$$

which ensure that the quantity of commodity assigned to each distribution center does not exceed its remaining availability.

4.3 Sequential solution approaches

In the previous two sections, we presented how the two subproblems, the SPC and the SPD, are solved. We now show how we combine the two approaches in order to obtain a solution method for the MC2DP. Different strategies are proposed, all based on the sequential solution of the two subproblems: the SPD and the SPC. In particular, we propose strategies where the SPD is solved first and the SPC second (indicated as SPD \rightarrow SPC), or, vice-versa, first we solve the SPC and then we solve the SPD (indicated as SPC \rightarrow SPD). In both cases, the solution of the first subproblem determines the quantity of each commodity at each distribution center. The second subproblem takes this information and deals with delivery or collection accordingly. Note that when the first subproblem is solved, the solution is such that the minimization of the transportation cost of that specific problem only is considered, regardless of the other subproblem. This may lead to solutions of poor quality for the MC2DP. As a consequence, we propose different strategies to guide the solution of the first subproblem and obtain better solutions of the MC2DP.

4.3.1 Sequential solution: SPD \rightarrow SPC

When the SPD is solved first, we consider three strategies to determine the values of quantities U_{dm}^C available at the distribution centers. The SPD is then solved through the algorithm presented in Section 4.2. Note that a solution of the SPD gives the required quantities U_{dm}^D of each commodity m at each distribution center d , computed as the sum of the demands of the *customer-commodities* assigned to this distribution center. Afterward, the SPC is solved by fixing the values of y variables and the required quantities at the depots U_{dm}^D in \mathcal{F}_{SPC} according to the solution of the SPD. The three strategies are as follows.

SPD infinite \rightarrow SPC. In this strategy, the values U_{dm}^C involved in the GAP formulation (Section 4.2.1) and in the ALNS algorithm (Section 4.2.2) are set to a valid upper bound as follows:

$$U_{dm}^C = \sum_{i \in \mathcal{V}_C} D_{im}, \quad \forall d \in \mathcal{V}_D, m \in \mathcal{M}$$

and do not restrict the search.

SPD finite balanced \rightarrow SPC. In this strategy, the values U_{dm}^C are restrictive and aim at balancing the quantity of commodities distributed from each distribution center. As a consequence, the quantity of each commodity available at each distribution center is determined as:

$$U_{dm}^C = \frac{\sum_{i \in \mathcal{V}_C} D_{im}}{|\mathcal{V}_D|} + \max_{i \in \mathcal{V}_C} \{D_{im}\}, \quad \forall d \in \mathcal{V}_D, m \in \mathcal{M}.$$

Note that the second term involved in the computation of U_{dm}^C guarantees that the SPD has a feasible solution and it also allows some flexibility in the assignment of *customer-commodities* to distribution centers.

SPD finite supplier based \rightarrow SPC. In this strategy, the values U_{dm}^C are determined taking into account the location of the distribution centers and the suppliers. The idea is to compute the available quantities U_{dm}^C at distribution center d based on the available quantities of the suppliers located close to d . First, we assign each supplier to its k closest distribution centers. Then, the quantity U_{dm}^C of commodity m available at distribution center d is computed as the sum of the available quantities O_{sm} for commodity m over all suppliers assigned to distribution center d . Note that if there are few distribution centers, k will take value 1, but when there are several distribution centers, k can be larger than 1 to consider larger values for U_{dm}^C , providing more flexibility for the solution of the SPD.

4.3.2 Sequential solution: SPC \rightarrow SPD

When the SPC is solved first, we consider three strategies. Based on the strategy chosen, we define different values for U_{dm}^D , and the formulation \mathcal{F}_{SPC} is solved to obtain a solution for the SPC. Then, in order to solve the SPD, the U_{dm}^C values are defined from the solution of \mathcal{F}_{SPC} as follows:

$$U_{dm}^C = \sum_{s \in \mathcal{V}_S} q_{sd}^m, \quad \forall d \in \mathcal{V}_D, m \in \mathcal{M}. \quad (38)$$

Finally, the SPD is solved. The three strategies considered when the SPC is solved first are listed in the following.

SPC not full truck \rightarrow SPD. In this strategy, the SPC is first solved using the model \mathcal{F}_{SPC} presented in Section 4.1 with values U_{dm}^D set to a non-binding value, i.e.:

$$U_{dm}^D = 0, \quad \forall d \in \mathcal{V}_D, m \in \mathcal{M}.$$

SPC full truck \rightarrow SPD. In this strategy, the SPC is first solved using the model \mathcal{F}_{SPC} presented in Section 4.1 with values U_{dm}^D set to 0 as in the former strategy. Then, the solution of the SPC is updated by using the remaining capacity of the trucks to increase the quantities q_{sd}^m brought to the distribution centers. This does not modify the cost of the SPC, but offers more flexibility when solving the SPD due to larger availabilities (U_{dm}^C) at distribution centers. More precisely, for each supplier s and each distribution center d , we divide the remaining capacity of the trucks traveling from s to d

by the number of commodities available at supplier s . This gives an equal maximum quantity q_{fill} for each commodity in order to fill the trucks. If the availability of some commodity is less than q_{fill} , we fill the trucks as much as we can and we possibly repeat the procedure. Details of the procedure are given in Algorithm 2 in the Appendix A. This procedure increases the values of variables q_{sd}^m in formulation \mathcal{F}_{SPC} . Then, in order to solve the SPD, U_{dm}^C values are defined as in Equation (38).

SPC full truck customer based \rightarrow SPD. This strategy solves the model \mathcal{F}_{SPC} in which quantities U_{dm}^D are determined on the basis of the locations of the customers. The idea is to compute the quantities U_{dm}^D required at distribution center d based on the demands of the customers located close to the distribution center d and far from other distribution centers. Customers close to several distribution centers are not included in the computation of U_{dm}^D to ensure some flexibility.

Given two distribution centers d_1 and d_2 with a distance a between d_1 and d_2 , we say that a customer i is d_1 - d_2 compatible if one of the following two conditions is satisfied: (1) the distance between i and d_1 is less than $a/3$; or (2) the distance between i and d_1 is less than a , and the distance between i and d_2 is greater than a . An example is given in Figure 4. For each distribution center d , we say that a customer i is assigned to d if for all other distribution centers $d' \in \mathcal{V}_D \setminus \{d\}$, i is d - d' compatible. With the choice of $a/3$, customers at similar distance from two distribution centers are not assigned to any distribution center, as illustrated in Figure 4.

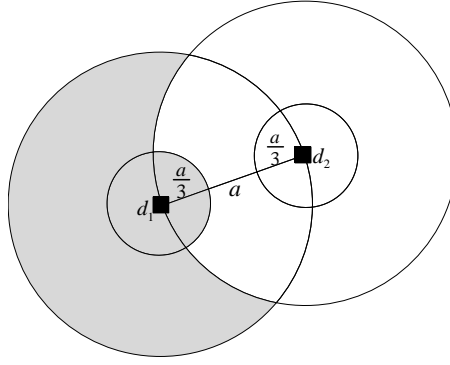


Figure 4: All customers in the grey zone are d_1 - d_2 compatible.

Then, the quantity U_{dm}^D is computed as the sum of the demands of commodity $m \in \mathcal{M}$ of customers assigned to $d \in \mathcal{V}_D$. After solving \mathcal{F}_{SPC} , the strategy *SPC full truck \rightarrow SPD* is applied to increase the quantities q_{sd}^m . Finally, values U_{dm}^C are calculated as in Equation (38).

5 Computational experiments

In this section, we compare the solution algorithms proposed in Section 4.3 to solve the MC2DP sequentially. An algorithm is determined by the sequence in which the two subproblems are solved and the strategy to solve the first subproblem.

In Section 5.1, we describe the generation of the sets of instances for the MC2DP based on instances for the C-SDVRP. Then, in Section 5.2, we report the results obtained when solving the instances with the different sequential algorithms. Then, we provide in Section 5.3 an analysis of the results to identify the instance characteristics that mostly impact the algorithm performance. Finally, in Section 5.4 we evaluate the quality of the solutions obtained with the sequential approach, with respect to the solving of the formulation \mathcal{F}_{MC2DP} with a commercial solver.

The algorithms have been implemented in C++ and run on an Intel (R) Core(TM) i7-4600U, 2.10GHz, and 16GB of RAM. We summarize the notation used to present the instances and the results in Table 1.

Symbol	Meaning
$nbIns$	Number of instances in each group
$SPDcost$	Best solution cost for the SPD
$SPCcost$	Best solution cost for the SPC
$Cost$	Total cost for the MC2DP
$avg.t(s)$	Average CPU time for computing the MC2DP solution (in seconds)
$avg.SP Dt(s)$	Average CPU time for computing the SPD solution (in seconds)
$avg.SP Ct(s)$	Average CPU time for computing the SPC solution (in seconds)

Table 1: Notation for computational results.

5.1 Instances

First, we create a base set of instances that is indicated as \mathcal{S} . Then, the other 11 sets are generated by modifying one of the characteristics of set \mathcal{S} .

5.1.1 Generation of the base set of instances \mathcal{S}

The delivery subproblem (SPD) of the MC2DP extends the C-SDVRP by considering multiple distribution centers. Hence, we build the base set of instances from instances of the C-SDVRP. More precisely, we consider the 64 small instances with 15 customers proposed by Archetti et al. (2014). These instances are built from the customer locations of the R101 and C101 Solomon instances for the VRP with Time Windows (Solomon, 1987). Each instance is characterized by five parameters listed thereafter. $I \in \{R101, C101\}$ indicates if the instance is based on R101 or C101. Only the depot and the first 15 customer locations are considered from the Solomon instances. The number of commodities, indicated by M , is equal to 2 or 3. A customer requires a commodity with probability p equal to 0.6 or 1. The quantity of each commodity required by a customer varies within the intervals $\Delta = [1, 100]$ or $\Delta = [40, 60]$. Last, parameter $\alpha \in \{1.1, 1.5, 2, 2.5\}$ determines the vehicle capacity by multiplying the vehicle capacity in the original Solomon instances. We indicate by $\mathcal{P} = (I, M, p, \Delta, \alpha)$ the set of parameters listed above.

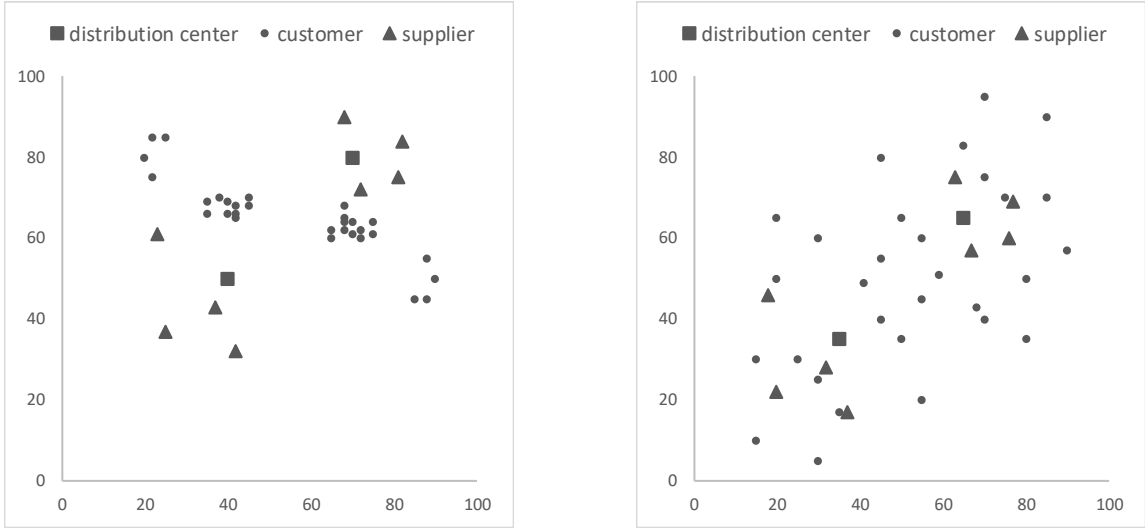
For each instance \mathcal{I} of the C-SDVRP, we create an instance \mathcal{I}_{MC2DP} for the MC2DP as follows. Given the coordinates (x_d, y_d) of the distribution center in \mathcal{I} , one distribution center of \mathcal{I}_{MC2DP} is located in (x_d, y_d) , while another one is located in $(x_d + \delta, y_d + \delta)$. Given the coordinates (x, y) of a customer in \mathcal{I} , one customer of \mathcal{I}_{MC2DP} is located in (x, y) , while another customer is located in $(-x + 2x_d + \delta, -y + 2y_d + \delta)$. Both customers have the same demand as in \mathcal{I} . Hence, the instance \mathcal{I}_{MC2DP} contains one distribution center d_1 and a set \mathcal{V}_C^1 of 15 customers with the same locations as instance \mathcal{I} . It contains as well another distribution center d_2 and a second set \mathcal{V}_C^2 of 15 customers. Customers in \mathcal{V}_C^2 are transposed by (δ, δ) from the original locations, and are also rotated by 180 degrees around the distribution center d_2 . After some preliminary experiments detailed in Appendix B, we set $\delta = 30$.

To locate the 8 suppliers we proceeded as follows. We consider a circle of radius r centered at each distribution center. We randomly created inside each circle four suppliers. Should the two circles intersect, the intersection is not considered as a potential zone to locate suppliers. The value of the radius r has been fixed to 30. Note that the cost c_{ij} associated with each arc $(i, j) \in \mathcal{A}$ is equal to the Euclidean distance between i and j .

The quantity of commodity $m \in \mathcal{M}$ available at each supplier is calculated as $O_{sm} = \lceil \zeta \cdot \frac{\sum_{i \in \mathcal{V}_C} D_{im}}{|\mathcal{V}_S|} \rceil$, where ζ is a parameter that has been fixed to 1.2. Hence, all suppliers produce all commodities with the same amount. Globally, the total supply is 20% more than the total demand. The motivation for this characteristic of the instances is to avoid feasibility problems and always obtain a solution of the heuristics. We wanted to focus on the coordination aspect of the two echelons rather than on feasibility issues. When $SPD \rightarrow SPC$, once the SPD is solved, the assignment of *customer-commodities* to distribution centers is known, that is the quantity of the commodity delivered to each distribution center is known. Then the SPC is optimally solved by using the mathematical model.

Thus, the 20% flexibility of the supply does not impact this heuristic. When $SPC \rightarrow SPD$, we should ensure that the solution of the SPC is feasible for the SPD. This is the reason why we allowed a larger than demand supply. However, note that having a larger supply than the total demand does not imply in practice a waste of products. Without being part of the system, the producers need to consecrate a consistent part of their working time to sell and deliver operations that are not their core business. The system would take care of a large segment of their production, freeing part of the working day. The remaining products may be sold across different channels, as on-site sales.

The capacities Q_S and Q_D of the trucks and vehicles are equal to the capacity of vehicles in \mathcal{I} . Figure 5 shows the locations of customers, distribution centers and suppliers for C101 and R101 configurations.



(a) Locations in instances obtained from C101. (b) Locations in instances obtained from R101.

Figure 5: Locations in instances in \mathcal{S} .

In order to ensure the diversity of the instances to evaluate the proposed sequential approach, we construct 11 additional sets of instances based on instances in \mathcal{S} . Each of the 11 sets differs from \mathcal{S} by the modification of one characteristic.

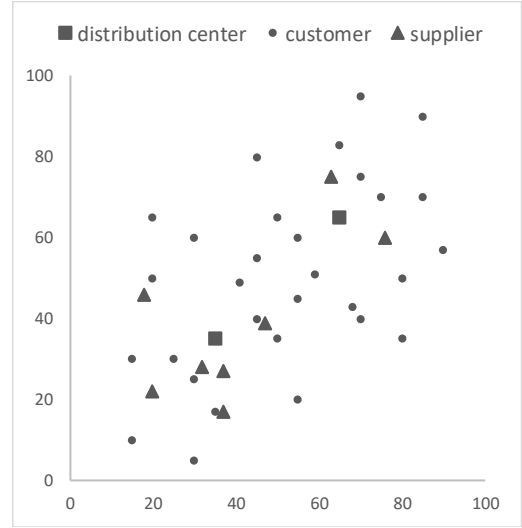
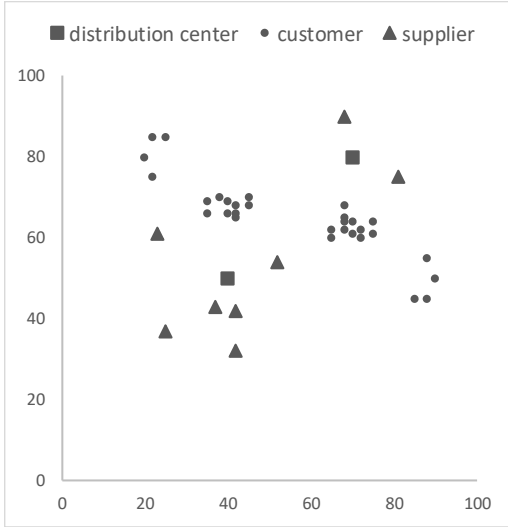
5.1.2 Modification of the supplier locations

In set \mathcal{S} , four suppliers are generated around each distribution center. We generate two other sets of instances where we modify the location of suppliers by unbalancing the number of suppliers generated around each distribution center. These two sets of instances are named \mathcal{S}_1^S and \mathcal{S}_2^S . As for instance set \mathcal{S} , we consider a circle of radius $r = 30$ centered at each distribution center, and no supplier is located in the intersection of the two circles. The characteristics of the new sets of instances are the following.

\mathcal{S}_1^S : Around distribution center d_1 we randomly locate 6 suppliers. The other two suppliers are randomly located around d_2 .

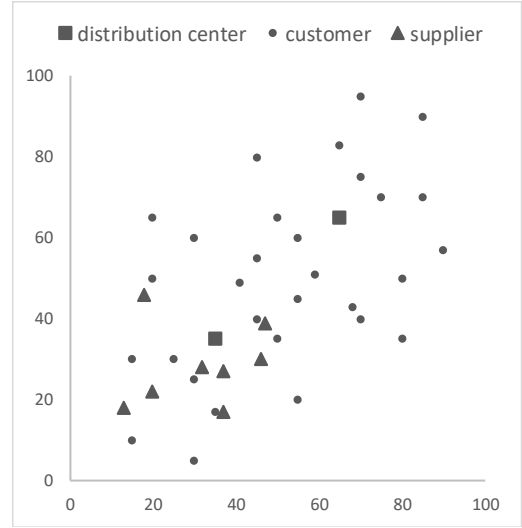
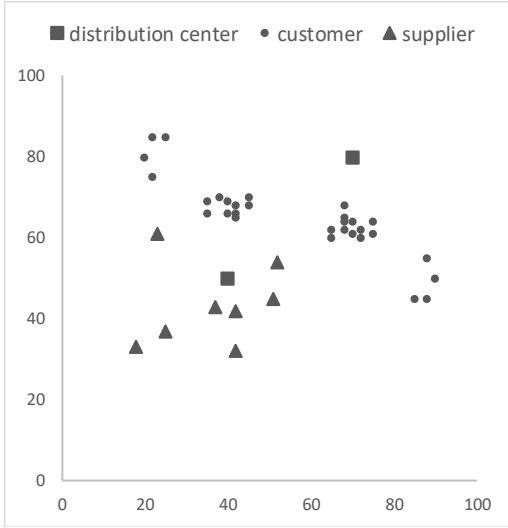
\mathcal{S}_2^S : Around distribution center d_1 we randomly locate 8 suppliers. No supplier is generated around d_2 .

Examples of the two configurations are presented in Figures 6 and 7.



(a) Locations in instances obtained from C101. (b) Locations in instances obtained from R101.

Figure 6: Locations in instances of \mathcal{S}_1^S .



(a) Locations in instances obtained from C101. (b) Locations in instances obtained from R101.

Figure 7: Locations in instances of \mathcal{S}_2^S .

5.1.3 Modification of the customer locations

In set \mathcal{S} , each distribution center is surrounded by 15 customers. This makes the instances *balanced* from the point of view of the customer locations with respect to the distribution centers. We generate four sets of instances where we modified the location of customers by unbalancing the number of customers around each distribution center. These four sets of instances are named \mathcal{S}_1^C , \mathcal{S}_2^C , \mathcal{S}_3^C and \mathcal{S}_4^C . As with instance set \mathcal{S} , given the coordinates (x, y) of a customer in \mathcal{I} , one customer of \mathcal{I}_{MC2DP} is located in (x, y) , while another customer is located in $(-x + 2x_d + \delta, -y + 2y_d + \delta)$. For instances in \mathcal{S} , the value $\delta = 30$ has been considered. For the new sets of instances, the locations of the 15 duplicated customers are defined as follows.

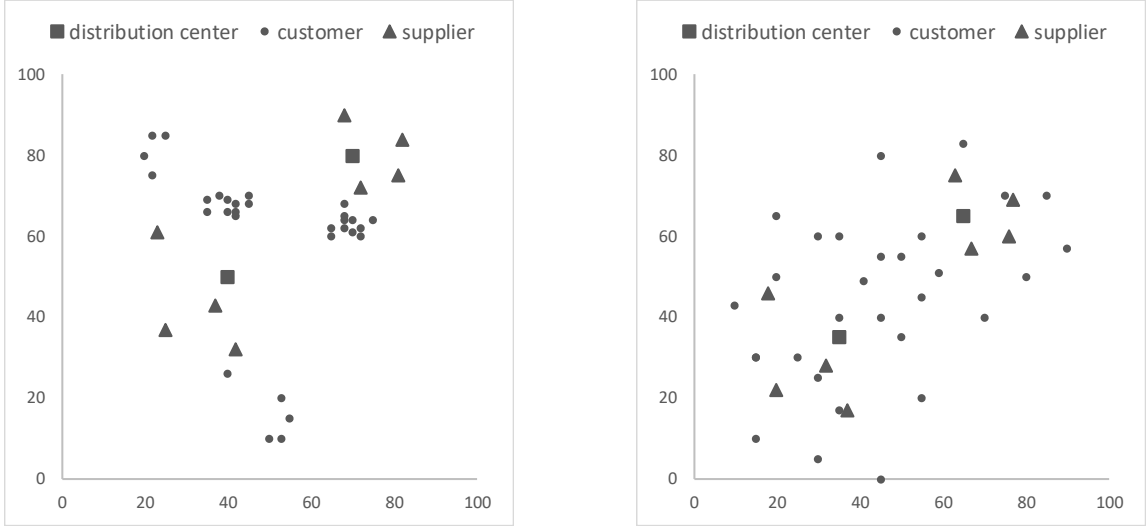
\mathcal{S}_1^C : For the first 5 customers in \mathcal{I} , $\delta = -5$; for the remaining 10 customers, $\delta = 30$.

\mathcal{S}_2^C : For the first 5 customers in \mathcal{I} , $\delta = 10$; for the remaining 10 customers, $\delta = 30$.

\mathcal{S}_3^C : For the first 10 customers in \mathcal{I} , $\delta = -5$; for the remaining 5 customers, $\delta = 30$.

\mathcal{S}_4^C : For the first 10 customers in \mathcal{I} , $\delta = 10$; for the remaining 5 customers, $\delta = 30$.

An illustration of instances in \mathcal{S}_1^C is given in Figure 8.



(a) Locations in instances obtained from C101. (b) Locations in instances obtained from R101.

Figure 8: Locations in instances of \mathcal{S}_1^C .

5.1.4 Modification of the available quantities at suppliers

In set \mathcal{S} , given a commodity $m \in \mathcal{M}$, all suppliers $s \in \mathcal{V}_S$ have the same available quantity $O_{sm} = \lceil \zeta \cdot \frac{\sum_{i \in \mathcal{V}_C} D_{im}}{|\mathcal{V}_S|} \rceil$, with $\zeta = 1.2$. We generate a set of instances \mathcal{S}^O where we unbalance the available quantities at suppliers. For this set of instances, we only consider the 32 instances where the number of commodities is $M=2$. Half of the suppliers, all located around the same distribution center, have $O_{s1} = \lceil \zeta \cdot \frac{1.8 \sum_{i \in \mathcal{V}_C} D_{i1}}{|\mathcal{V}_S|} \rceil$ units of commodity 1 and $O_{s2} = \lceil \zeta \cdot \frac{0.2 \sum_{i \in \mathcal{V}_C} D_{i2}}{|\mathcal{V}_S|} \rceil$ units of commodity 2. On the contrary, the other suppliers, all located around the other distribution center, have $O_{s1} = \lceil \zeta \cdot \frac{0.2 \sum_{i \in \mathcal{V}_C} D_{i1}}{|\mathcal{V}_S|} \rceil$ units of commodity 1 and $O_{s2} = \lceil \zeta \cdot \frac{1.8 \sum_{i \in \mathcal{V}_C} D_{i2}}{|\mathcal{V}_S|} \rceil$ units of commodity 2. ζ is still fixed to 1.2.

5.1.5 Modification of the number of suppliers

The instances in \mathcal{S} all contain 8 suppliers. Thus, we create other two sets of instances, indicated with $\mathcal{S}_1^{S_{add}}$ and $\mathcal{S}_2^{S_{add}}$, that contain 10 and 12 suppliers, respectively. The number of customers is kept as in the base set \mathcal{S} , i.e., equal to 30 for all instances. The new added suppliers are randomly placed around the two distribution centers, such that the same number of suppliers are placed around each distribution center.

5.1.6 Modification of the number of customers

The instances in \mathcal{S} all contain 30 customers. We then create two sets of instances, indicated with $\mathcal{S}_1^{C_{add}}$ and $\mathcal{S}_2^{C_{add}}$ where the number of customers is increased to 50 and 70 respectively. The number of suppliers is kept as in \mathcal{S} , i.e., equal to 8.

To locate the new customers, we randomly select customers in the original instance and we relocate them by translating the coordinates by a value that is randomly chosen in $[-20, 20]$. The demand of the new customer is equal to the demand of the original customer. The quantity of each commodity available at each supplier is determined as described in Section 5.1.1.

To sum up, in this work, we generate 12 sets of instances. Each set contains 64 instances except for set \mathcal{S}^O , which has 32 instances.

5.2 Comparison of the sequential heuristics to solve the MC2DP

The six heuristics presented in Section 4.3 for the solution of the MC2DP are tested on the 12 instance sets presented in Section 5.1. The mathematical programs presented in Section 4.2 are solved using Cplex 12.6. The ALNS algorithm is run with a limit of 5000 iterations and the values of the parameters used in Gu et al. (2019). The additional parameters introduced in Section 4.2.2 are set to $E_{inf} = 50$ and $E_{feas} = 50$ after preliminary experiments. We recall that the values of γ_{min} and λ_{min} equal the cost of the initial solution. Moreover, for the algorithm *SPD finite supplier based* \rightarrow *SPC*, we set $k = 1$ since the instances contain few distribution centers.

In order to compare the algorithms, for each instance set we report:

- $\Delta avg.SPCcost$: percentage difference between the average cost for collection ($avg.SPCcost$) and the minimum average cost for collection ($minavg.SPCcost$), i.e. $avg.\Delta_{SPCcost} = 100 * (avg.SPCcost - minavg.SPCcost) / minavg.SPCcost$;
- $\Delta avg.SPDCost$: percentage difference between the average cost for delivery ($avg.SPDCost$) and the minimum average cost for delivery ($minavg.SPDCost$), i.e. $avg.\Delta_{SPDCost} = 100 * (avg.SPDCost - minavg.SPDCost) / minavg.SPDCost$;
- $\Delta avg.Cost$: percentage difference between the average total cost ($avg.Cost$) and the minimum average total cost ($minavg.Cost$), i.e. $\Delta avg.Cost = 100 * (avg.Cost - minavg.Cost) / minavg.Cost$.

We also report the minimum average results for the collection cost, the delivery cost and the total cost in the last line of each table.

Table 2 reports the results for the instance set \mathcal{S} . The best value obtained for the MC2DP, the SPC and the SPD are provided in bold. The table shows that the best algorithm is *SPD finite balanced* \rightarrow *SPC*. The other algorithms that solve first the SPD provide also good results, always better than the algorithms where the SPC is solved first. The worst algorithm is *SPC not full truck* \rightarrow *SPD*, with an average increase in total cost of 2.35%. We also note that, when solving the SPD first, we have an increase in the cost of the SPC of more than 8%. Instead, when solving first the SPC, the increase on the SPD cost goes from 4.49% to 7.54%. However, the algorithms solving thhe SPD first provide better values of total cost. This is due to the unbalance in the total cost of the SPC and the SPD. Finally, it is interesting to note that the algorithm providing the best total cost is not the best for none of the two subproblems. On average, the CPU times are around 1 minute. Since computational times on the other sets ($\mathcal{S}_1^S, \mathcal{S}_2^S, \mathcal{S}_1^C, \mathcal{S}_2^C, \mathcal{S}_3^C, \mathcal{S}_4^C, \mathcal{S}^O$) of instances are similar, we do not report them in the following Tables 3, 4, 5 and 6.

Strategy	$\Delta avg.SPCcost$	$\Delta avg.SPDCost$	$\Delta avg.Cost$	$avg.SPCT(s)$	$avg.SPDT(s)$	$avg.t(s)$
<i>SPD infinite</i> \rightarrow <i>SPC</i>	8.60	0.01	0.02	0.15	46.84	46.99
<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	8.55	0.01	0.00	0.15	54.27	54.42
<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	8.71	0.00	0.04	0.16	50.59	50.75
<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	7.54	2.35	0.26	81.23	81.49
<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	4.74	0.49	0.27	73.21	73.47
<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	0.00	4.49	0.33	0.27	74.96	75.23
minimum average costs	331.92	711.99	1072.34	-	-	-

Table 2: Average results on instance set \mathcal{S} .

Results on instance sets \mathcal{S}_1^S and \mathcal{S}_2^S with modification of the supplier locations are reported in Table 3. We note that, among the three algorithms where the SPD is solved first, the algorithm

SPD finite supplier based \rightarrow *SPC* provides a high cost for the SPD. However this is compensated by a low cost for the SPC resulting in the lowest cost for the whole problem. When supplier locations around the distribution centers are unbalanced, the algorithm *SPD finite supplier based* \rightarrow *SPC* is very efficient in comparison with the other algorithms that solve first the SPD. We also note that, for these sets of instances, the sequence in which the subproblems are solved has a significant impact with respect to the instances in the base set \mathcal{S} , providing a larger increase in the cost of the second subproblem, especially on the set \mathcal{S}_2^S . The best algorithm is *SPD finite supplier based* \rightarrow *SPC*. On \mathcal{S}_2^S , *SPC not full truck* \rightarrow *SPD* has the same performance as *SPD finite supplier based* \rightarrow *SPC*. However, we note that the remaining algorithms in which the SPC is solved first perform better than the remaining algorithms where the SPD is solved first, especially on \mathcal{S}_2^S .

Instance set	\mathcal{S}_1^S			\mathcal{S}_2^S		
	$\Delta avg.SPCcost$	$\Delta avg.SPDCost$	$\Delta avg.Cost$	$\Delta avg.SPCcost$	$\Delta avg.SPDCost$	$\Delta avg.Cost$
<i>SPD infinite</i> \rightarrow <i>SPC</i>	45.36	0.00	6.55	100.57	0.00	9.78
<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	45.05	0.00	6.45	98.33	0.00	9.15
<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	3.94	9.01	0.00	0.00	34.86	0.00
<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	17.57	4.27	0.00	34.86	0.00
<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	13.44	1.65	0.00	34.89	0.01
<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	7.01	9.04	0.93	42.52	12.62	0.14
minimum average costs	331.92	712.04	1121.17	376.85	712.08	1337.16

Table 3: Average results on instance sets \mathcal{S}_1^S and \mathcal{S}_2^S .

Results on sets \mathcal{S}_1^C , \mathcal{S}_2^C , \mathcal{S}_3^C , and \mathcal{S}_4^C are reported in Tables 4 and 5. These instances have unbalanced customer locations around the distribution centers. For these four sets, the algorithms *SPD finite balanced* \rightarrow *SPC* and *SPD supplier based* \rightarrow *SPC* provide the best results for the whole problem. The total costs achieved with these two algorithms are very similar. It can be noticed that when applying the algorithm *SPD infinite* \rightarrow *SPC*, the results are not competitive with the two other algorithms where the SPD is solved first: the cost of the SPD is smaller but the cost of the SPC increases a lot, leading to a high value of the total cost. Moreover, for the three algorithms where the SPC is solved first, we observe a similar behaviour for these sets of instances and the base set \mathcal{S} . The only difference is that the algorithm *SPC full truck customer based* \rightarrow *SPD* provides the best results among the three algorithms where the SPC is solved first. Note that when the SPC is solved first and the customer locations are not taken into account (algorithms *SPC not full truck* \rightarrow *SPD* and *SPC full truck* \rightarrow *SPD*), the solutions that are obtained for the SPC are always the same and have the lowest SPC cost for all the considered instances. However, the overall solution is never the best. This clearly shows the importance of taking into account the location of final customers when bringing commodities to distribution centers.

Instance set	\mathcal{S}_1^C			\mathcal{S}_2^C		
	$\Delta avg.SPCcost$	$\Delta avg.SPDCost$	$\Delta avg.Cost$	$\Delta avg.SPCcost$	$\Delta avg.SPDCost$	$\Delta avg.Cost$
<i>SPD infinite</i> \rightarrow <i>SPC</i>	28.61	0.00	3.24	25.47	0.00	2.78
<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	11.75	2.85	0.00	11.77	2.30	0.04
<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	14.78	1.48	0.03	13.64	1.35	0.00
<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	14.62	4.05	0.00	12.82	3.15
<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	10.11	1.14	0.00	8.42	0.33
<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	0.84	8.85	0.58	0.23	8.24	0.29
minimum average costs	331.92	710.99	1102.21	331.92	690.85	1077.35

Table 4: Average results on instance sets \mathcal{S}_1^C and \mathcal{S}_2^C .

Table 6 presents the results obtained on set \mathcal{S}^O where the quantities available at the suppliers are unbalanced. The results indicate that in this case, the algorithm *SPC full truck customer based* \rightarrow *SPD* provides the lowest total cost. Hence, when the suppliers have unbalanced available quantities,

Instance set	\mathcal{S}_3^C			\mathcal{S}_4^C		
	$\Delta avg.SPCcost$	$\Delta avg.SPDCost$	$\Delta avg.Cost$	$\Delta avg.SPCcost$	$\Delta avg.SPDCost$	$\Delta avg.Cost$
<i>SPD infinite</i> \rightarrow <i>SPC</i>	65.70	0.00	9.63	55.71	0.00	8.95
<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	12.46	9.85	0.46	11.87	6.84	0.00
<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	15.79	7.56	0.00	15.30	5.22	0.03
<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	23.99	5.67	0.00	18.89	3.90
<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	19.44	2.84	0.00	14.63	1.26
<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	10.20	12.15	1.25	0.72	13.58	0.82
minimum average costs	331.92	717.99	1156.61	331.92	684.00	1102.15

Table 5: Average results on instance sets \mathcal{S}_3^C and \mathcal{S}_4^C .

starting by solving the SPC in a sequential approach is the best choice.

Strategy	$\Delta avg.SPCcost$	$\Delta avg.SPDCost$	$\Delta avg.Cost$
<i>SPD infinite</i> \rightarrow <i>SPC</i>	112.24	0.00	9.06
<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	113.50	0.03	9.40
<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	4.25	38.77	2.09
<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	45.59	4.68
<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	43.39	3.49
<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	40.16	17.74	0.00
minimum average costs	345.76	722.98	1335.83

Table 6: Average results on instance set \mathcal{S}^O .

The results on sets of instances $\mathcal{S}_1^{S_{add}}$, $\mathcal{S}_2^{S_{add}}$, $\mathcal{S}_1^{C_{add}}$ and $\mathcal{S}_2^{C_{add}}$ are reported in Tables 7, 8, 9 and 10, respectively.

Strategy	$\Delta avg.SPCcost$	$\Delta avg.SPDCost$	$\Delta avg.Cost$	$avg.SPCt(s)$	$avg.SPDCt(s)$	$avg.t(s)$
<i>SPD infinite</i> \rightarrow <i>SPC</i>	10.70	0.01	0.23	0.04	52.45	52.48
<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	10.18	0.00	0.08	0.06	93.26	93.32
<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	10.90	0.00	0.29	0.06	85.91	85.97
<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	8.29	2.75	0.45	134.97	135.43
<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	4.54	0.20	0.46	107.30	107.76
<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	0.00	4.25	0.00	0.33	139.04	139.36
minimum average costs	304.89	712.03	1047.15	-	-	-

Table 7: Average results on instance set $\mathcal{S}_1^{S_{add}}$.

It can be observed that the results obtained by the different sequential heuristics are similar to the results obtained on the instance set \mathcal{S} . In particular, the best results are generally obtained when the SPD is solved first. It can also be noted the the worst strategy is, in all cases, *SPC not full truck* \rightarrow *SPD*, as it is the case for the instance set \mathcal{S} . We can observe that the computational time increases with respect to the number of customers, while it is little impacted by the addition of suppliers. This can be explained by the greater difficulty in solving a larger SPD, where routes have to be determined, compared with solving a larger SPC, where direct trips only are considered.

Results on the different sets of randomly generated instances reveal that *SPD finite supplier based* \rightarrow *SPC* and *SPC full truck customer based* \rightarrow *SPD* are the best algorithms. As a rule, *SPD finite supplier based* \rightarrow *SPC* identifies the best solutions in most cases except when the quantities available at the suppliers are unbalanced. In the latter case, *SPD finite supplier based* \rightarrow *SPC* has a better performance.

5.3 Analysis of the performance of the algorithms

In this section, we present an analysis of the results presented in Section 5.2 to identify the instance characteristics that mostly impact the algorithm performance. More specifically, we aim at finding

Strategy	$\Delta avg.SPCcost$	$\Delta avg.SPDCost$	$\Delta avg.Cost$	$avg.SPCT(s)$	$avg.SPDT(s)$	$avg.t(s)$
<i>SPD infinite</i> \rightarrow <i>SPC</i>	4.74	0.01	0.00	0.08	85.48	85.56
<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	4.86	0.00	0.03	0.07	93.44	93.51
<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	4.78	0.00	0.01	0.07	88.58	88.65
<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	2.99	0.64	0.35	120.08	120.42
<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	2.08	0.02	0.37	111.97	112.34
<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	0.00	2.18	0.09	0.39	113.24	113.63
minimum average costs	308.52	712.03	1035.21	-	-	-

Table 8: Average results on instance set $\mathcal{S}_2^{S_{add}}$.

Strategy	$\Delta avg.SPCcost$	$\Delta avg.SPDCost$	$\Delta avg.Cost$	$avg.SPCT(s)$	$avg.SPDT(s)$	$avg.t(s)$
<i>SPD infinite</i> \rightarrow <i>SPC</i>	4.92	0.00	0.00	0.06	156.93	156.99
<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	4.91	0.22	0.11	0.05	162.13	162.18
<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	4.17	1.38	0.42	0.07	165.33	165.39
<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	13.53	5.25	0.29	266.26	266.55
<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	10.53	3.61	0.29	226.27	226.57
<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	0.64	5.59	1.19	0.30	219.97	220.27
minimum average costs	863.88	1087.28	1993.67	-	-	-

Table 9: Average results on instance set $\mathcal{S}_1^{C_{add}}$.

Strategy	$\Delta avg.SPCcost$	$\Delta avg.SPDCost$	$\Delta avg.Cost$	$avg.SPCT(s)$	$avg.SPDT(s)$	$avg.t(s)$
<i>SPD infinite</i> \rightarrow <i>SPC</i>	7.09	0.00	0.00	0.20	617.93	618.13
<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	7.82	0.33	0.50	0.20	654.17	654.37
<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	2.79	6.84	1.60	0.18	573.09	573.27
<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	12.06	3.03	0.75	1031.30	1032.04
<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	9.66	1.79	0.76	853.73	854.49
<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	2.02	4.51	0.04	0.32	462.36	462.68
minimum average costs	1384.20	1583.68	3066.04	-	-	-

Table 10: Average results on instance set $\mathcal{S}_2^{C_{add}}$.

rules to determine the best solution approach, among the ones we proposed, based on the instance characteristics. This is of crucial importance for the decision-maker. In fact, instance characteristics are known before determining a solution for the problem.

Starting from the results presented in Table 2, we already noticed that all algorithms in which the SPD is solved first perform better than the ones in which the SPC is solved first, even though the gap is not substantial. We note that the instances on which these results were obtained, set \mathcal{S} , are well balanced in terms of location of customers and suppliers: two distribution centers are present, and customers and suppliers are evenly distributed in the neighborhood of the two. In such a case, the two sequential schemes tend to give similar solutions, in terms of assignment of customers and suppliers to distribution centers. We explain that all algorithms in which the SPD is solved first produce better solutions than when the SPC is solved first by the fact that the major cost component in the instances is the routing cost. The same is true in Tables 7–10, where instances are still balanced.

From Tables 3–6 we see, instead, that when instances are unbalanced in terms of location of suppliers and/or customers or in terms of quantity requested/available, then it is important to take into account this unbalance when choosing the most appropriate solution method. The best strategy is the one that optimizes first the unbalanced part, i.e., the SPD first when customers are unbalanced and the SPC first when suppliers are unbalanced. In addition, the results also show that, in this case, it is important to consider a strategy that links the two subproblems. In fact, *SPD finite supplier based* \rightarrow *SPC* and *SPC full truck customer based* \rightarrow *SPD* are the two more robust strategies. Even though each of them is not always the best strategy among those solving the SPD and the SPC first, respectively, their gap with respect to the best solution is always small and has lower variations with respect to the other strategies.

A summary of the results in Tables 3–10 is reported in Table 11. We see that the maximum gap of *SPD finite supplier based* \rightarrow *SPC* is 2.09% while the one of *SPC full truck customer based* \rightarrow *SPD* is 1.25%. All other strategies present a gap that, in the worst case, is well above 3%.

Strategy	\mathcal{S}_1^S	\mathcal{S}_2^S	\mathcal{S}_1^C	\mathcal{S}_2^C	\mathcal{S}_3^C	\mathcal{S}_4^C	\mathcal{S}^O	$\mathcal{S}_1^{S_{add}}$	$\mathcal{S}_2^{S_{add}}$	$\mathcal{S}_1^{C_{add}}$	$\mathcal{S}_2^{C_{add}}$
<i>SPD infinite</i> \rightarrow <i>SPC</i>	6.55	9.78	3.24	2.78	9.63	8.95	9.06	0.23	0.00	0.00	0.00
<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	6.45	9.15	0.00	0.04	0.46	0.00	9.40	0.08	0.03	0.11	0.50
<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	0.00	0.00	0.03	0.00	0.00	0.03	2.09	0.29	0.01	0.42	1.60
<i>SPC not full truck</i> \rightarrow <i>SPD</i>	4.27	0.00	4.05	3.15	5.67	3.90	4.68	2.75	0.64	5.25	3.03
<i>SPC full truck</i> \rightarrow <i>SPD</i>	1.65	0.01	1.14	0.33	2.84	1.26	3.49	0.20	0.02	3.61	1.79
<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	0.93	0.14	0.58	0.29	1.25	0.82	0.00	0.00	0.09	1.19	0.04

Table 11: Summary of $\Delta_{avg.Cost}$ from Tables 3–10 .

5.4 Evaluation of the sequential heuristics

This section is devoted to the evaluation of the quality of the solutions obtained by the sequential schemes. To this end, we created four sets of small instances. The new instances have 10, 15, 20 and 25 customers, respectively. Instances with 10 to 20 customers have 4 suppliers while instances with 25 customers have 6 suppliers. Each set contains 9 instances.

To create these instances we randomly selected 3 instances from \mathcal{S} , \mathcal{S}_1^S and \mathcal{S}_2^S . The original instances are all from the class $C101$ with $\alpha = 1.1$ and $\Delta = [1, 100]$. From the selected instance we randomly removed customers and suppliers.

The compact formulation \mathcal{F}_{MC2DP} for the MC2DP, presented in Section 2.1, as well as the sequential heuristics to solve the MC2DP are evaluated over these instances. Formulation \mathcal{F}_{MC2DP} is solved using Cplex 12.8. A maximum computational time of 1 hour is given to Cplex.

The results are reported in Table 12. The first three columns of the table give the characteristics of the instance, where M indicates the number of commodities and p is the probability that a customer requires one commodity, as previously explained (see Section 5.1). Column *set* indicates the instance

set from which the instance is created. The fourth, fifth and sixth columns report results obtained by solving formulation \mathcal{F}_{MC2DP} with Cplex 12.8. Column UB reports the value of the best feasible solution provided by Cplex, while Column LB reports the best lower bound. The symbol \times indicates that Cplex was not able to provide a feasible solution. Column $t(s)/gap(\%)$ is a hybrid column that reports the computational time, in seconds, when Cplex solves the instance to optimality in the given amount of time or the optimality percentage gap, computed as $100 \cdot \frac{UB-LB}{LB}$. The last three columns report the performance of the sequential heuristics. Column $best$ reports the value of the best solution obtained over the six sequential heuristics presented in Section 4.3, column $t(s)$ reports the total time, in seconds, needed to run the six heuristics. The last column provides the percentage gap between the value of the solution obtained by Cplex and the value of the solution obtained by the sequential approaches. This is computed as $100 \cdot \frac{best-UB}{UB}$. Whenever columns UB and $best$ report the same value, a symbol $*$ is used to indicate that the sequential approaches succeed in finding the optimal solution, while the symbol \checkmark indicates that the sequential approaches found the same solution as Cplex, but optimality has not been proved.

Instance			Cplex			Seq		
M	p	set	UB	LB	$t(s)/gap(\%)$	best	$t(s)$	gap
$N_C = 10$								
2	0.6	\mathcal{S}	394.66	394.62	240	394.66	50	*
2	1	\mathcal{S}	579.52	549.03	5.55%	579.52	58	\checkmark
3	0.6	\mathcal{S}	470.77	470.73	378	470.77	59	*
2	0.6	\mathcal{S}_1^S	406.52	406.49	108	406.52	48	*
2	1	\mathcal{S}_1^S	562.34	562.28	1441	562.34	57	*
3	0.6	\mathcal{S}_1^S	437.98	437.94	486	437.98	70	*
2	0.6	\mathcal{S}_2^S	406.52	406.48	57	406.52	41	*
2	1	\mathcal{S}_2^S	663.52	663.46	2261	663.52	52	*
3	0.6	\mathcal{S}_2^S	463.58	463.53	41	463.58	58	*
average Cplex/ALNS gap								0.00%
$N_C = 15$								
2	0.6	\mathcal{S}	512.50	472.59	8.45%	512.50	89	\checkmark
2	1	\mathcal{S}	771.34	654.28	17.89%	765.47	133	-0.76%
3	0.6	\mathcal{S}	561.62	525.54	6.87%	551.88	122	-1.73%
2	0.6	\mathcal{S}_1^S	533.43	468.67	13.82%	533.43	96	\checkmark
2	1	\mathcal{S}_1^S	790.28	682.49	15.79%	784.05	134	-0.79%
3	0.6	\mathcal{S}_1^S	573.69	476.41	20.42%	554.73	152	-3.30%
2	0.6	\mathcal{S}_2^S	590.55	550.62	7.25%	590.55	85	\checkmark
2	1	\mathcal{S}_2^S	896.12	742.96	20.62%	893.09	133	-0.34%
3	0.6	\mathcal{S}_2^S	590.71	512.49	15.26%	590.71	122	\checkmark
average Cplex/ALNS gap								-0.77%
$N_C = 20$								
2	0.6	\mathcal{S}	638.30	522.83	22.09%	636.71	270	-0.25%
2	1	\mathcal{S}	1063.71	807.35	31.75%	1007.04	295	-5.63%
3	0.6	\mathcal{S}	719.37	561.35	28.15%	708.62	376	-1.52%
2	0.6	\mathcal{S}_1^S	705.02	512.40	37.59%	668.50	255	-5.46%
2	1	\mathcal{S}_1^S	1108.31	851.34	30.18%	1077.43	292	-2.87%
3	0.6	\mathcal{S}_1^S	787.84	571.23	37.92%	775.30	152	-1.62%
2	0.6	\mathcal{S}_2^S	713.38	557.91	27.87%	713.16	236	-0.03%
2	1	\mathcal{S}_2^S	1243.96	867.64	43.37%	1177.46	253	-5.65%
3	0.6	\mathcal{S}_2^S	846.41	615.72	37.47%	835.00	311	-1.37%
average Cplex/ALNS gap								-2.71%
$N_C = 25$								
2	0.6	\mathcal{S}	\times	621.30		818.57	141	
2	1	\mathcal{S}	1188.39	904.92	31.32%	1184.62	173	-0.32%
3	0.6	\mathcal{S}	1049.31	661.20	58.70%	827.46	168	-26.81%
2	0.6	\mathcal{S}_1^S	872.00	633.00	37.76%	849.29	138	-2.67%
2	1	\mathcal{S}_1^S	1416.33	915.13	54.77%	1258.91	164	-12.50%
3	0.6	\mathcal{S}_1^S	1209.09	643.48	87.90%	894.56	164	-35.16%
2	0.6	\mathcal{S}_2^S	\times	648.72		881.02	175	
2	1	\mathcal{S}_2^S	1526.08	952.69	60.19%	1367.61	147	-11.59%
3	0.6	\mathcal{S}_2^S	1100.07	662.89	65.95%	939.52	151	-17.09%
average Cplex/ALNS gap								-15.16%

Table 12: Comparison of the sequential approaches with results obtain from Cplex on \mathcal{F}_{MC2DP} (in 1 hour computation).

As it can be seen from Table 12, on instances with 10 customers the sequential approaches can always find the optimal solution whenever Cplex is able to prove optimality. When the number of customers goes up to 20 or 25, the sequential approaches always find a better solution, with an average improvement of more than 15% for instances with 25 customers. As expected, the lower bounds that Cplex provides are too poor to draw any conclusions on the quality of the solutions that the sequential approach obtains. The only conclusion that can be drawn from the analysis of Table 12 is that, as the size of the instance increases, a heuristic approach is the only viable method, since Cplex begins to struggle to provide feasible solutions. On tests run with 30 customers and not reported in the table, Cplex identified a feasible solution for one instance only. In addition, we carried out additional tests by letting Cplex run for two hours instead of one, but this led to a very minor improvement of the lower bound. Thus, the corresponding results are not reported.

6 A case study

In this section, we present a study on instances generated from a real case application. The aim is to show how the sequential solution approaches behave on real-case applications.

6.1 Context

The case study was proposed by local authorities of the French department of Isère (General Council and Chamber of Agriculture). The aim is to increase the volume of fresh food products sold through short and local supply chains in the department of Isère. The department of Isère is an interesting location for fresh food supply chains since: i) there are many farmers producing a variety of fruits and vegetables, ii) these farmers have very low revenues when selling their products through classical distribution channels, iii) the inhabitants of the department are deeply interested in buying local fruits and vegetables through short and local supply chains.

Local authorities of Isère identified two kinds of customers for short and local supply chains: school canteens and supermarkets. For such customers, considering direct deliveries from the farmers is not acceptable as farmers should devote a significant part of their working time to carrying out deliveries. Hence, the idea is to use a set of distribution centers which would be jointly managed by associations of farmers.

6.2 Description of the data sets

In this case study, the commodities are fresh fruits and vegetables: apple, pear, kiwi, strawberry, carrot, lettuce, tomato, zucchini, cucumber, potato. In order to reflect the seasonality of these products, demand and supply are not constant over the year. Hence, we consider a set of patterns for demand and supply. Each pattern gives rise to one instance whose objective is to capture the availability of the products of the month.

Two sets of customers are considered: school canteens and supermarkets. A study conducted with the General Council of Isère has permitted to collect and estimate actual demands for school canteens. The demand associated with supermarkets has instead been generated according to the following procedure. The selected supermarkets are the ones selling fresh food products with a sales area of at least $300m^2$. Demands for each supermarket have been generated under the following hypotheses: i) the demands for each product has the same proportion as for school canteens (i.e. people have the same kind of food consumption at home and in school canteens), ii) demands are proportional to the sales area of the supermarket, and iii) the global demand of all commodities is proportional to the global revenue generated by the sales of the considered fresh products through supermarkets, for which the data are provided by a statistical study from the General Council of

Isère. For school canteens, 8 patterns of demands and supply are considered, and for supermarkets 10 patterns are considered. Hence, there is a set of 18 instances for the case study.

The instances are divided in two sets: one considering canteens and one considering supermarkets. This is because the General Council of Isère is interested in evaluating the possibility of implementing a network that supplies school canteens or supermarkets from producers using distribution centers. However, the possibility of having a shared network to serve these two sets of customers is not taken into account. This is mainly due to the different demand of products in the two cases, with the supermarket demands being much more important. The effect of this difference is twofold. First, when school canteens are involved, producers could use their vehicle to bring commodities to the distribution centers since the daily demand is limited. On the other hand, when supermarkets are considered this is not anymore a viable option since producers would spend an important part of their working time performing round trips to the distribution centers. In this case, it is supposed that farmers may wish to purchase larger vehicles as a counterpart of participating in the collaborative system. Second, to deliver to supermarkets one can suppose that large trucks can be used efficiently due to the volume of the demand. In addition, dedicated docks are usually available in supermarkets to unload vehicles. On the other side, the demand of canteens is smaller and not sufficient to fill such large trucks, which are also unlikely to be able to access them. Due to these reasons, the General Council of Isère wishes to study the two cases separately in order to consider the possible implementation of one network.

In the school canteens instances, there are 103 customers and up to 61 suppliers. For each pattern of supply and demand, the number of commodities ranges from 5 to 8, and the number of suppliers producing at least one commodity ranges from 54 to 61. In the supermarket instances, there are 188 customers and up to 61 suppliers. For each pattern of supply and demand, the number of commodities ranges from 5 to 8, and the number of suppliers producing at least one commodity ranges from 45 to 61. In all instances, there are 5 distribution centers. Their locations have been proposed by the General Council and the Chamber of Agriculture of Isère. The sizes of the instances are reported in Tables 13 and 14. The locations of customers, suppliers and distribution centers in the department of Isère are shown in Figure 9. The capacity of the vehicles is set such that $Q_D = 2Q_S$, since farmers in short and local supply chains generally do not have large capacity vehicles while distribution centers may invest in vehicles with bigger capacity to visit several customers.

instance id	$ \mathcal{V}_C $	$ \mathcal{V}_S $	$ \mathcal{V}_D $	$ \mathcal{M} $
1	103	61	5	8
2	103	61	5	7
3	103	61	5	6
4	103	61	5	5
5	103	61	5	6
6	103	61	5	5
7	103	61	5	6
8	103	54	5	7

Table 13: School canteens instances.

6.3 Analysis of the results

The instances are very large with respect to the ones described in Section 5.1 since there are up to 1500 *customer-commodities*. The computational time of the ALNS significantly increases with the size of the instances. Hence, when solving the SPD, the ALNS algorithm is run with a limit of 100 iterations. Moreover, due to the size of the instances, we set a time limit for solving mixed integer linear programs. More specifically, we set a 5 minutes time limit to solve the GAP model

instance id	$ \mathcal{V}_C $	$ \mathcal{V}_S $	$ \mathcal{V}_D $	$ \mathcal{M} $
1	188	61	5	6
2	188	61	5	5
3	188	61	5	6
4	188	54	5	7
5	188	45	5	5
6	188	58	5	6
7	188	61	5	8
8	188	61	5	7
9	188	61	5	7
10	188	61	5	6

Table 14: Supermarkets instances.

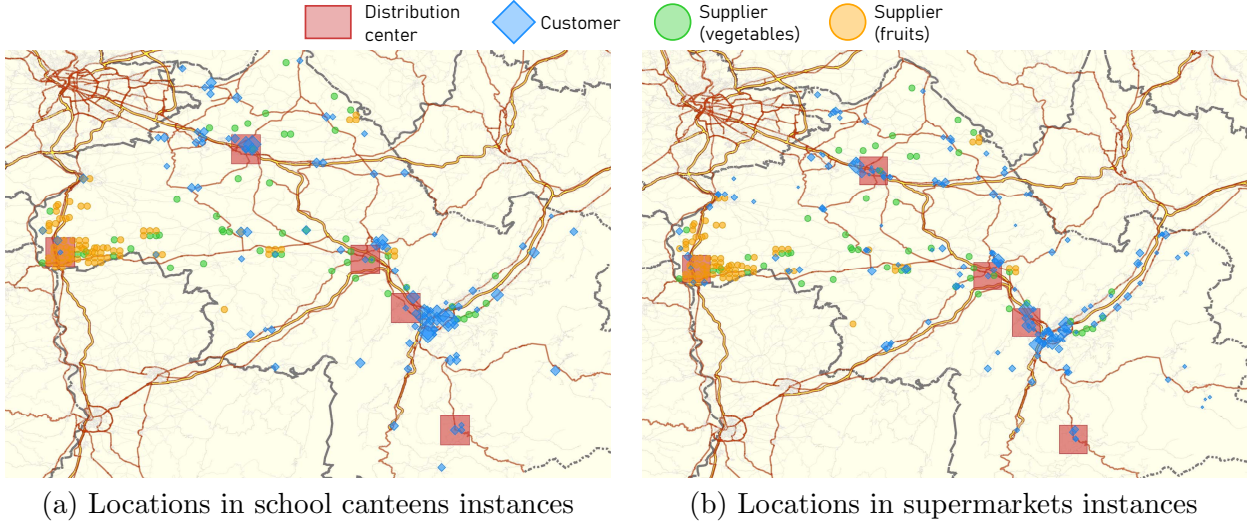


Figure 9: Locations in the case study instances.

presented in Section 4.2.1 to obtain an initial solution of the SPD, and a 10 minutes time limit to solve the formulation \mathcal{F}_{SPC} . As there are 5 distribution centers, when applying the algorithm *SPD finite supplier based* \rightarrow *SPC*, we set $k = 2$, i.e., each supplier is assigned to its two nearest distribution centers.

The results obtained for each sequential approach are reported in Table 15 for the school canteens and in Table 16 for the supermarkets. The columns nbT and nbV indicate the numbers of trucks used for collection operations and the number of vehicles used for delivery operations, respectively. The columns $\Delta SPCcost$, $\Delta SPDcost$ and $\Delta Cost$ report the percentage difference between the SPC cost and the minimum SPC cost, the SPD cost and the minimum SPD cost, the total cost and the minimum total cost, respectively.

When customers are school canteens, two algorithms outperform the others, namely: *SPC full truck* \rightarrow *SPD* and *SPD finite supplier based* \rightarrow *SPC*. It is interesting to note that *SPC full truck* \rightarrow *SPD* provides the best results on average. However, this algorithm gives low collection costs and high delivery costs. This is an interesting observation: in fact, local authorities might judge as inappropriate a solution where the cost is highly unbalanced, even if it provides the best cost for the overall system. On the contrary, the algorithm *SPD finite supplier based* \rightarrow *SPC* is a bit more costly on average, but has the advantage of balancing the costs between collection and delivery. It can also be noticed that the algorithms *SPD finite balanced* \rightarrow *SPC* and *SPC full truck customer based* \rightarrow *SPD* do not provide good solutions. This could be explained by the fact that suppliers and customers are not located homogeneously around the distribution centers.

instance id	Strategy	$\Delta SPCcost$	$\Delta SPDcost$	$\Delta Cost$	$SPCt(s)$	$SPDt(s)$	$t(s)$	nbT	nbV
1	<i>SPD infinite</i> \rightarrow <i>SPC</i>	144.24	0.00	9.76	9.57	683.38	693.41	45	15
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	144.73	6.62	11.96	219.67	677.39	898.69	45	16
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	78.58	38.27	0.40	4.94	1256.32	1262.17	44	15
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	174.72	17.23	7.42	4763.64	4771.05	41	15
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	119.01	0.00	6.81	1981.80	1988.61	41	16
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	18.42	121.12	6.60	46.42	2463.34	2509.75	41	15
	minimum costs	928.66	889.69	2877.15	-	-	-	-	-
2	<i>SPD infinite</i> \rightarrow <i>SPC</i>	143.59	0.00	13.71	36.51	397.44	434.31	45	14
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	155.49	8.91	20.54	118.95	490.81	611.13	45	14
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	73.51	41.52	3.24	4.23	844.34	849.31	42	14
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	188.16	24.87	7.30	2883.20	2890.50	41	16
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	109.56	0.00	6.62	1576.24	1582.87	41	14
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	15.80	104.43	3.70	10.53	1562.47	1573.00	42	15
	minimum costs	928.66	872.17	2756.37	-	-	-	-	-
3	<i>SPD infinite</i> \rightarrow <i>SPC</i>	199.86	0.00	11.48	2.56	373.58	376.46	32	8
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	245.65	1.43	23.07	3.57	270.39	275.09	35	9
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	108.46	26.80	0.00	1.80	671.94	674.45	33	9
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	166.24	28.66	3.03	2324.59	2327.62	30	11
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	93.72	0.17	3.09	1518.40	1521.50	30	9
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	46.56	92.17	10.78	6.93	1575.55	1582.48	30	10
	minimum costs	498.48	813.24	2070.29	-	-	-	-	-
4	<i>SPD infinite</i> \rightarrow <i>SPC</i>	169.70	0.00	16.60	3.00	312.69	315.97	36	8
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	199.26	2.33	26.31	4.44	192.39	197.71	38	9
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	98.44	29.40	5.66	2.52	471.45	474.94	37	8
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	153.19	20.14	3.19	1887.38	1890.56	35	8
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	96.58	0.00	3.22	787.23	790.45	35	8
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	36.04	87.90	7.73	6.30	942.31	948.61	35	9
	minimum costs	644.11	763.10	2144.25	-	-	-	-	-
5	<i>SPD infinite</i> \rightarrow <i>SPC</i>	133.83	0.00	19.27	22.02	301.71	324.09	49	12
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	126.99	6.84	18.50	6.87	380.67	388.94	50	12
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	61.94	37.71	1.70	6.93	596.74	604.61	49	13
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	189.16	19.75	3.86	2227.06	2230.91	46	12
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	118.38	0.00	3.80	1272.64	1276.44	46	11
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	20.44	121.19	8.77	6.86	1327.52	1334.38	46	12
	minimum costs	1135.42	810.68	2905.79	-	-	-	-	-
6	<i>SPD infinite</i> \rightarrow <i>SPC</i>	168.15	0.00	28.23	4.58	262.93	267.80	41	9
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	160.32	6.14	27.50	7.85	142.78	151.62	42	9
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	69.19	34.51	4.66	2.89	345.44	348.85	42	7
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	175.68	27.12	3.06	1520.67	1523.72	38	8
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	94.15	0.00	3.11	931.38	934.48	38	8
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	24.68	103.07	11.71	5.31	738.85	744.16	38	9
	minimum costs	790.98	742.95	2233.42	-	-	-	-	-
7	<i>SPD infinite</i> \rightarrow <i>SPC</i>	164.61	0.00	21.63	1.70	447.42	449.44	40	9
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	147.56	1.22	16.34	4.61	287.01	292.67	42	9
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	73.91	26.23	0.00	3.56	861.23	865.45	41	10
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	167.72	22.18	4.27	1938.72	1942.99	38	11
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	112.05	3.73	4.37	1570.88	1575.25	38	9
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	32.95	89.43	7.25	9.25	1793.77	1803.02	38	8
	minimum costs	790.98	784.04	2365.34	-	-	-	-	-
8	<i>SPD infinite</i> \rightarrow <i>SPC</i>	101.10	0.00	11.30	16.75	552.88	569.94	37	11
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	111.95	5.87	17.49	2.34	371.81	375.29	36	9
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	68.41	15.25	5.24	0.89	779.23	780.76	36	9
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	172.27	40.32	3.37	3942.81	3946.19	34	10
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	66.68	0.00	3.34	1570.57	1573.91	34	9
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	19.16	77.19	10.98	5.91	1494.92	1500.83	34	11
	minimum costs	727.33	763.74	2000.32	-	-	-	-	-

Table 15: Detailed results for school canteens instances.

For the instances with supermarkets, two algorithms outperform the others, namely: *SPD finite supplier based* \rightarrow *SPC* and *SPD infinite* \rightarrow *SPC*. *SPD finite supplier based* \rightarrow *SPC* provides the best results on 7 instances out of 10, and has the advantage of balancing the costs between collection and delivery operations. It can be noticed that some sequential algorithms lead to a very high cost, namely *SPD finite balanced* \rightarrow *SPC* and *SPC not full truck* \rightarrow *SPD*.

We note that the algorithm *SPD finite supplier based* \rightarrow *SPC* again provides very good results on these additional sets of instances. However, depending on the instance setting, some approaches based on the SPC solution first can be competitive.

From a computational point of view, solving the SPD first takes much less time than solving the SPC first. Moreover, most of the time is spent on solving the SPD. When solving the SPC first, the available quantities at the distribution centers are very limited, which may generate a significant infeasibility when applying destroy and repair moves in the ALNS algorithm. This makes it more difficult to obtain good solutions with the local search.

instance id	Strategy	$\Delta SPCcost$	$\Delta SPDcost$	$\Delta Cost$	$SPCt(s)$	$SPDt(s)$	$t(s)$	nbT	nbV
1	<i>SPD infinite</i> \rightarrow <i>SPC</i>	214.00	0.00	3.48	611.18	1820.26	2432.16	77	36
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	245.69	20.25	16.42	606.65	2482.62	3091.35	76	36
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	108.98	93.16	0.00	51.29	2713.73	2767.41	76	35
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	265.70	15.04	6.97	8391.10	8398.06	77	36
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	234.38	7.34	6.97	6459.01	6465.98	77	37
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	10.69	224.58	7.62	15.73	10286.01	10301.74	76	36
	minimum costs	1620.35	1585.66	6449.01	-	-	-	-	-
2	<i>SPD infinite</i> \rightarrow <i>SPC</i>	228.00	0.00	4.24	605.75	1277.51	1883.88	75	31
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	268.59	24.63	20.04	605.28	2116.40	2723.28	73	32
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	125.48	88.55	0.00	40.10	2066.67	2108.86	73	31
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	250.33	7.44	4.89	5610.75	5615.64	71	33
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	222.78	0.92	5.05	4205.70	4210.75	71	30
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	11.45	226.89	4.71	10.20	5946.75	5956.95	71	34
	minimum costs	1519.26	1462.82	6183.81	-	-	-	-	-
3	<i>SPD infinite</i> \rightarrow <i>SPC</i>	205.51	0.00	0.00	610.75	1931.09	2542.59	73	34
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	252.42	23.57	17.42	612.11	2658.14	3272.19	73	35
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	116.08	90.33	0.76	33.61	2905.92	2940.85	72	34
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	242.07	10.40	6.78	8684.81	8691.59	71	36
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	224.71	6.04	7.28	6999.63	7006.91	71	37
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	16.20	208.08	5.84	15.25	9249.36	9264.61	71	36
	minimum costs	1512.12	1548.85	6168.55	-	-	-	-	-
4	<i>SPD infinite</i> \rightarrow <i>SPC</i>	96.70	0.00	3.88	608.26	2972.43	3581.45	86	37
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	116.11	22.50	18.04	603.28	4000.20	4605.78	85	38
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	48.34	45.47	0.00	155.32	3876.63	4033.38	84	37
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	231.68	39.65	8.27	14438.65	14446.91	84	38
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	160.36	17.59	8.42	10476.51	10484.93	84	37
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	11.55	163.04	22.71	22.65	12376.38	12399.03	84	40
	minimum costs	1985.06	1654.69	5351.62	-	-	-	-	-
5	<i>SPD infinite</i> \rightarrow <i>SPC</i>	72.48	0.00	4.77	605.59	1257.91	1864.09	77	34
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	110.85	21.16	27.70	604.33	1563.74	2169.92	77	34
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	47.10	16.78	0.00	604.88	1559.48	2165.50	77	33
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	178.09	35.17	7.99	5495.13	5503.12	74	34
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	102.42	9.58	7.95	4345.60	4353.55	74	36
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	4.93	102.26	11.56	12.16	4415.38	4427.54	74	38
	minimum costs	1816.33	1493.05	4415.45	-	-	-	-	-
6	<i>SPD infinite</i> \rightarrow <i>SPC</i>	154.81	0.00	1.63	606.33	1554.77	2161.85	98	48
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	193.37	22.16	18.83	606.05	2318.84	2927.02	98	44
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	85.71	72.96	0.00	175.93	2385.50	2562.88	95	43
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	238.19	17.34	10.68	8621.81	8632.49	96	45
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	193.98	5.89	10.54	6007.16	6017.70	96	47
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	9.58	187.48	7.05	18.90	7724.43	7743.33	96	46
	minimum costs	2091.43	1823.58	7037.98	-	-	-	-	-
7	<i>SPD infinite</i> \rightarrow <i>SPC</i>	162.14	0.00	3.96	605.93	2970.73	3577.50	109	51
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	199.68	22.26	20.81	605.41	6528.79	7136.93	109	50
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	80.68	83.98	0.00	610.58	6411.95	7024.80	108	49
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	258.95	18.69	14.14	16299.01	16313.14	107	49
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	214.87	7.83	14.03	20570.22	20584.25	107	53
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	7.90	265.78	22.76	66.09	16548.57	16614.66	106	53
	minimum costs	2374.80	1933.26	7847.62	-	-	-	-	-
8	<i>SPD infinite</i> \rightarrow <i>SPC</i>	170.10	0.00	0.00	605.34	2497.02	3103.21	105	48
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	206.99	25.41	16.43	607.10	5173.62	5783.09	107	46
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	104.89	81.77	0.84	604.99	4574.01	5181.14	105	46
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	258.53	12.85	17.90	10955.42	10973.32	102	48
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	220.44	3.87	17.56	8705.85	8723.41	102	52
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	8.34	194.36	0.08	20.73	12724.18	12744.91	101	51
	minimum costs	2250.39	1876.13	7954.41	-	-	-	-	-
9	<i>SPD infinite</i> \rightarrow <i>SPC</i>	197.45	0.00	0.00	605.49	2294.35	2900.67	93	44
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	235.90	27.01	16.25	606.94	3506.61	4115.86	91	45
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	126.24	83.97	1.48	605.33	3923.75	4534.08	93	41
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	265.42	11.78	9.30	12015.01	12024.31	87	42
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	239.52	5.68	9.30	11836.48	11845.78	87	45
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	11.59	218.23	3.65	17.14	9850.93	9868.07	87	45
	minimum costs	1911.03	1751.77	7436.06	-	-	-	-	-
10	<i>SPD infinite</i> \rightarrow <i>SPC</i>	214.00	0.00	3.48	611.83	1812.64	2425.15	77	36
	<i>SPD finite balanced</i> \rightarrow <i>SPC</i>	246.19	20.25	16.55	606.66	2481.68	3090.51	76	36
	<i>SPD finite supplier based</i> \rightarrow <i>SPC</i>	108.98	93.16	0.00	55.61	2827.97	2887.82	76	35
	<i>SPC not full truck</i> \rightarrow <i>SPD</i>	0.00	245.62	10.11	6.88	8623.59	8630.47	77	36
	<i>SPC full truck</i> \rightarrow <i>SPD</i>	0.00	234.38	7.34	6.94	6372.42	6379.35	77	37
	<i>SPC full truck customer based</i> \rightarrow <i>SPD</i>	10.69	239.23	11.22	15.91	8250.89	8266.80	76	37
	minimum costs	1620.35	1585.66	6449.01	-	-	-	-	-

Table 16: Detailed results for supermarkets.

When customers are school canteens, few trucks are used. Their number is less than the number of suppliers. This means that some suppliers do not provide any commodity since they are far from each distribution center. When customers are supermarkets, the total supply is not very large compared to the total demand. Hence, more trucks are used for the delivery and several suppliers perform more than one trip to a distribution center.

6.4 On the benefit of consolidation

As explained in the previous section, delivery operations to school canteens and supermarkets should be kept separated due to different vehicles used to perform the delivery. However, collection operations may be consolidated and carried out only once for both school canteens and supermarkets.

To study this scenario, we created 5 new instances from the instances presented in the previous section. These new instances are obtained by merging one school canteen and one supermarket instance characterized by the same number of commodities, the same number of suppliers, and the same distance matrix with respect to suppliers and distribution centers. It should be noted that the distance between a canteen and a supermarket is assumed to be infinite. Indeed, the two sets of canteens and supermarket were considered separately, and we only have a distance matrix for each set. Hence, we were not able to recompute distances between the two sets of customers.

In order to keep the sequential nature of the solution approach, we tested three strategies and compared the results with total cost obtained by simply summing up the costs obtained by solving the two instances separately.

The three strategies are:

- solve the SPD for canteens and supermarkets, determine the quantity of each commodity required at the distribution centers by summing up what is needed in each case, then solve the SPC with this requirement. This strategy is indicated by $SPD \rightarrow SPC$;
- simultaneously solve the SPC for canteens and supermarkets, then solve the SPD for canteens, update the available quantity at distribution centers, then solve the SPD for supermarkets. This strategy is indicated by $SPC \rightarrow SPDCanteens1st$;
- simultaneously solve the SPC for canteens and supermarkets, then solve the SPD for supermarkets, update the available quantity at distribution centers, then solve the SPD for canteens. This strategy is indicated by $SPC \rightarrow SPDSupermarkets1st$

Each time an instance of the SPC or the SPD is solved, the three corresponding strategies, defined in Section 4.3, are run in order to solve the supproblem, and the best result is kept. The results are reported in Tables 17-18. A dash means that the approach did not find a feasible solution for the corresponding instance. In Table 17, column *Base* reports the costs for the separated case. It is calculated as the sum of the costs of the two original instances corresponding to the merged instances, i.e., the sum of the costs of the canteen and supermarket instances. Columns $SPD \rightarrow SPC$, $SPC \rightarrow SPDCanteens1st$, and $SPC \rightarrow SPDSupermarkets1st$ report the costs for the three proposed strategies. In Table 18, we report the evolution in percentage, of the costs of the three strategies, with respect to the cost of the separated case. As an example, the value 9.71 means that for the instance *merged1*, when using the strategy $SPD \rightarrow SPC$, the cost for the SPC increases by 9.71% with respect to the cost for the SPC in the separated case.

Instance	Base			$SPD \rightarrow SPC$			$SPC \rightarrow SPDCanteens1st$			$SPC \rightarrow SPDSupermarkets1st$		
	SPCcost	SPDcost	Cost	SPCcost	SPDcost	Cost	SPCcost	SPDcost	Cost	SPCcost	SPDcost	Cost
merged1	5949.19	4787.02	10736.21	6526.92	2812.63	9339.55	2339.38	-	-	2339.38	5819.71	8159.09
merged2	3513.41	7305.67	10819.08	6229.80	2770.40	9000.20	2216.94	-	-	2216.94	5259.22	7476.16
merged3	4306.44	3979.17	8285.61	4564.18	2337.58	6901.77	1433.72	-	-	1433.72	3928.00	5361.72
merged4	2163.38	6221.77	8385.15	4945.08	2184.41	7129.49	1486.66	-	-	1486.66	3844.91	5331.57
merged5	4169.47	3287.29	7456.76	2623.92	3510.20	6134.12	1930.55	-	-	1930.55	4419.69	6350.24

Table 17: Comparison of the results between separated and merged cases: solution cost.

As expected, consolidation of the collection reduces, in all the cases, the total cost of the solution with respect to the *Base* case. These reduction in total cost are important since the total cost is decreased by 13% to 36%. This cost reduction is often due to a very important cost reduction in the first subproblem solved.

Instance	<i>SPD</i> \rightarrow <i>SPC</i>			<i>SPC</i> \rightarrow <i>SPDCanteens1st</i>			<i>SPC</i> \rightarrow <i>SPDSupermarkets1st</i>		
	Δ SPCcost	Δ SPDcost	Δ Cost	Δ SPCcost	Δ SPDcost	Δ Cost	Δ SPCcost	Δ SPDcost	Δ Cost
merged1	9.71	-41.24	-13.01	-60.68	-	-	-60.68	21.57	-24.00
merged2	77.31	-62.08	-16.81	-36.90	-	-	-36.90	-28.01	-30.90
merged3	5.98	-41.25	-16.70	-66.71	-	-	-66.71	-1.29	-35.29
merged4	128.58	-64.89	-14.97	-31.28	-	-	-31.28	-38.20	-36.42
merged5	-37.07	6.78	-17.74	-53.70	-	-	-53.70	34.45	-14.84

Table 18: Comparison of the results between separated and merged cases: percentage difference for the merged cases with respect to the separated case (*Base*).

From Tables 17 and 18 one can observe that when the SPC is solved first, solving the SPD for school canteens before solving the SPD for the supermarkets brings to infeasibility. This is explained by the higher demands of supermarkets and the reduced flexibility that is left to the optimization after solving the SPD for school canteens.

We can also note that, in most of the cases, the best results are obtained when the SPC is solved first. In this case, larger savings are obtained from merging collection activities.

7 Conclusions and future research

In this paper, we presented a new and complex problem which occurs, for example, in local agri-food supply chains, the MC2DP (Multi-Commodity two-echelon Distribution Problem). The problem concerns the collection of commodities from suppliers to distribution centers and the delivery from distribution centers to final customers. The objective is to jointly optimize the transportation plan for the collection and delivery operations. In order to tackle this complex problem, we proposed two sequential approaches based on the decomposition of the problem in two subproblems: collection and delivery. For each sequential approach, three strategies are considered in order to take into account, in the first subproblem solved, the impact of its solution on the second subproblem solved.

The proposed algorithms have been compared on several sets of instances derived from instances from the literature. The best algorithm usually does not provide the lowest cost neither for the collection nor for the delivery subproblem. The algorithms are also tested on a case study of a short and local fresh food supply chain. Large size instances are considered with two types of customers, school canteens and supermarkets. We compared the different algorithms and provided some managerial insights about cost balancing between collection and delivery.

The MC2DP is a complex problem to solve. The proposed sequential approaches have the advantage of being easy to design and to be understood by decision makers. However, it is well known that sequential solution approaches may provide suboptimal solutions. Thus, the main future research direction is to develop an integrated solution approach solving the MC2DP as a whole instead of decomposing it in subproblems.

Future research will study the system in a multi-period horizon, allowing to take into account fairness aspects. In practical settings, the offer of products is larger than the demand. This implies that not all the producers may be involved in the supply of final customers during a given day. Conditions might be imposed on the quantity collected from producers on a multi-day basis, thus guaranteeing a fair treatment of each farmer.

Acknowledgments

This work is partially supported by supported by the CSC (China Scholarship Council) and by the ELSAT 2020 project. The authors thank *Conseil Général de l'Isère* and *Chambre d'Agriculture de l'Isère* for providing the case study instances.

References

- Agustina, D., Lee, C., and Piplani, R. (2014). Vehicle scheduling and routing at a cross docking center for food supply chains. *International Journal of Production Economics*, 152:29–41.
- Archetti, C., Campbell, A. M., and Speranza, M. G. (2014). Multicommodity vs. single-commodity routing. *Transportation Science*, 50(2):461–472.
- Beasley, J. E. (1983). Route first—cluster second methods for vehicle routing. *Omega*, 11(4):403–408.
- Berti, G. and Mulligan, C. (2016). Competitiveness of small farms and innovative food supply chains: The role of food hubs in creating sustainable regional and local food systems. *Sustainability*, 8(7):616.
- Blanquart, C., Gonçalves, A., Vandenbossche, L., Kebir, L., Petit, C., and Traversac, J.-B. (2010). The logistic leverages of short food supply chains performance in terms of sustainability. In *12th World Conference on Transport Research*.
- Bosona, T. G. and Gebresenbet, G. (2011). Cluster building and logistics network integration of local food supply chain. *Biosystems engineering*, 108(4):293–302.
- Bruzzzone, A., Massei, M., and Bocca, E. (2009). Fresh-food supply chain. In Merkuriev, Y., Merkurieva, G., Piera, M., and Guasch, A., editors, *Simulation-Based Case Studies in Logistics*, chapter 8, pages 127–146. Springer.
- Cattrysse, D. G. and Van Wassenhove, L. N. (1992). A survey of algorithms for the generalized assignment problem. *European Journal of Operational Research*, 60(3):260–272.
- Cuda, R., Guastaroba, G., and Speranza, M. G. (2015). A survey on two-echelon routing problems. *Computers & Operations Research*, 55:185–199.
- Dellaert, N., Van Woensel, T., Crainic, T. G., and Saridarq, F. D. (2021). A multi-commodity two-echelon capacitated vehicle routing problem with time windows: Model formulations and solution approach. *Computers & Operations Research*, 127:105154.
- Drexel, M. (2012). Synchronization in vehicle routing—a survey of vrps with multiple synchronization constraints. *Transportation Science*, 46(3):297–316.
- Flores, H. and Villalobos, J. R. (2018). A modeling framework for the strategic design of local fresh-food systems. *Agricultural Systems*, 161:1–15.
- Gu, W., Cattaruzza, D., Ogier, M., and Semet, F. (2019). Adaptive large neighborhood search for the commodity constrained split delivery VRP. *Computers & Operations Research*, 112:104761.
- Guastaroba, G., Speranza, M. G., and Vigo, D. (2016). Intermediate facilities in freight transportation planning: a survey. *Transportation Science*, 50(3):763–789.
- Guimarães, T. A., Coelho, L. C., Schenekemberg, C. M., and Scarpin, C. T. (2019). The two-echelon multi-depot inventory-routing problem. *Computers & Operations Research*, 101:220–233.
- Hinojosa, Y., Puerto, J., and F.R., F. (2000). A multiperiod two-echelon multicommodity capacitated plant location problem. *European Journal of Operational Research*, 123(2):271–291.
- King, R. P., Hand, M. S., and Gómez, M. I. (2015). *Growing Local: Case Studies on Local Food Supply Chains*. U of Nebraska Press.
- Ladier, A.-L. and Alpan, G. (2016). Cross-docking operations: Current research versus industry practice. *Omega*, 62:145–162.

- Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416.
- Laporte, G., Ropke, S., and Vidal, T. (2014). Heuristics for the vehicle routing problem. In Toth, P. and Vigo, D., editors, *Vehicle Routing: Problems, Methods, and Applications*, chapter 4, pages 87–116. MOS-SIAM Series on Optimization.
- Magnanti, T. L. and Wong, R. T. (1984). Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55.
- Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., and Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79:115 – 129.
- Ogier, M., Cung, V.-D., and Boissière, J. (2013). Service network design in short and local fresh food supply chain. *RAIRO-Operations Research*, 47(4):445–464.
- Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002.
- Qiu, Y., Zhou, D., Du, Y., Liu, J., Pardalos, P. M., and Qiao, J. (2021). The two-echelon production routing problem with cross-docking satellites. *Transportation Research Part E: Logistics and Transportation Review*, 147:102210.
- Rahbari, A., Mahdi Nasiri, M., Werner, F., Musavi, M., and Jolai, F. (2019). The vehicle routing and scheduling problem with cross-docking for perishable products under uncertainty: Two robust bi-objective models. *Applied Mathematical Modelling*, 70:605–625.
- Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472.
- Ross, G. T. and Soland, R. M. (1975). A branch and bound algorithm for the generalized assignment problem. *Mathematical Programming*, 8(1):91–103.
- Rucabado-Palomar, T. and Cuéllar-Padilla, M. (2018). Short food supply chains for local food: a difficult path. *Renewable Agriculture and Food Systems*, page 1–10.
- Sadjady, H. and Davoudpour, H. (2012). Two-echelon, multi-commodity supply chain network design with mode selection, lead-times and inventory costs. *Computer & Operations Research*, 39(7):1345–1354.
- Solomon, M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265.
- Van Belle, J., Valckenaers, P., and Cattrysse, D. (2012). Cross-docking: State of the art. *Omega*, 40:827–846.
- Vidal, T., Crainic, T., Gendreau, M., and Prins, C. (2013). Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, 231(1):1–21.
- Yu, S., Puchinger, J., and Sun, S. (2020). Two-echelon urban deliveries using autonomous vehicles. *Transportation Research Part E: Logistics and Transportation Review*, 141:102018.
- Yu, Y. and Xiao, T. (2021). Analysis of cold-chain service outsourcing modes in a fresh agri-product supply chain. *Transportation Research Part E: Logistics and Transportation Review*, 148:102264.
- Zhou, L., Baldacci, R., Vigo, D., and Wang, X. (2018). A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. *European Journal of Operational Research*, 265(2):765–778.

A Algorithm for the full truck strategy

Algorithm 2 Full truck strategy.

```

1: for each supplier  $s \in \mathcal{V}_S$  do
2:   for each commodity  $m \in \mathcal{M}$  do
3:     Compute the remaining quantity  $R_{sm} = O_{sm} - \sum_{d \in \mathcal{V}_D} q_{sd}^m$ 
4:   end for
5:   Let  $nbR_s$  be the number of commodities of supplier  $s$  such that  $R_{sm} > 0$ 
6: end for
7: for each distribution center  $d \in \mathcal{V}_D$  do
8:   for each supplier  $s \in \mathcal{V}_S$  do
9:     Compute the remaining capacity in the trucks  $R_{sd} = x_{sd} \times Q_S - \sum_{m \in \mathcal{M}} q_{sd}^m$ 
10:    while  $nbR_s > 0$  and  $R_{sd} > 0$  do
11:      Compute the average remaining capacity per commodity as  $avgR_{sd} = R_{sd}/nbR_s$ 
12:      for each commodity  $m \in \mathcal{M}$  do
13:        Determine the maximum quantity of commodity  $m$  that supplier  $s$  can insert in the
        trucks sent to DC  $d$  as  $Q_{sd}^m = \min\{avgR_{sd}, R_{sm}\}$ 
14:        Add  $Q_{sd}^m$  to  $q_{sd}^m$ 
15:        Remove  $Q_{sd}^m$  units from  $R_{sd}$ 
16:        Remove  $Q_{sd}^m$  units from  $R_{sm}$ 
17:      end for
18:      Update  $nbR_s$ 
19:    end while
20:   end for
21: end for

```

B Detailed results for setting the value of δ

We discuss here how we set the value of δ that determines the location of duplicated customers and distribution center. We solve the delivery subproblem (SPD) for each instance \mathcal{I}_{MC2DP} in the base set \mathcal{S} . The collection subproblem is not considered, and the SPD is solved as discussed in Section 4.2. The ALNS is run with a limit of 5000 iterations. To evaluate the solutions, we compare the cost obtained for the SPD ($SPDcost$) with twice the optimal value OPT for the corresponding instance of the C-SDVRP, as reported in Archetti et al. (2014). By construction of the instances, $2OPT$ is a valid upper bound for the SPD. These tests are performed with values of δ : 100, 50, 30, 20, 10, and 5.

The results are presented in Table 19. We present results for each group of instances, where a group is defined by a triplet (I, M, p) . Results for a group are averaged on the values of α and Δ . Each instance contains 30 customers and 2 distribution centers. Table 19 reports the average number of *customer-commodities* in each group of instances ($avg.n_{cc}$), the number of instances in the group ($nbIns$), the average deviation ($avg.\Delta$) of the cost obtained by solving the SPD and the upper bound $2OPT$, the average CPU time in seconds ($avg.t(s)$). Moreover, the last three columns respectively report the number of solutions such that the cost obtained by solving the SPD is higher than, equal to or lower than $2OPT$.

We observe that when δ is large (i.e., equal to 100), the solution value equals $2OPT$ most of the time. In this case, by construction of the instances, this value is most likely to be the optimal value. On the other side, when δ is smaller than 100, the value $2OPT$ is a valid upper bound that can be improved by running the algorithm for the SPD. When δ equals 5 or 10, for all instances we obtain a value lower than $2OPT$ since the distribution centers are then very close. We are interested in generating a set of instances where the average deviation from $2OPT$ is negative, but with some instances such that the cost of the SPD equals $2OPT$. Hence, we set $\delta = 30$.

Case 1: $\delta=100$

\mathcal{I}	M	p	$avg.n_{cc}$	$nbIns$	$avg.\Delta$	$avg.t(s)$	$SPDcost > 2OPT$	$SPDcost = 2OPT$	$SPDcost < 2OPT$
C101	2	0.6	44	8	0.00	33.21	0	8	0
C101	2	1	60	8	0.01	45.55	1	7	0
C101	3	0.6	56	8	0.00	52.80	0	8	0
C101	3	1	90	8	0.20	82.53	4	4	0
R101	2	0.6	44	8	0.00	32.53	0	8	0
R101	2	1	60	8	0.00	52.62	0	8	0
R101	3	0.6	56	8	0.00	49.43	0	8	0
R101	3	1	90	8	0.00	95.92	0	8	0
total				64	0.03	55.57	5	59	0

Case 2: $\delta=50$

C101	2	0.6	44	8	0.00	32.70	0	8	0
C101	2	1	60	8	0.00	47.42	0	8	0
C101	3	0.6	56	8	0.00	48.98	0	8	0
C101	3	1	90	8	0.32	81.94	4	4	0
R101	2	0.6	44	8	-1.02	28.02	0	3	5
R101	2	1	60	8	-0.40	48.23	0	5	3
R101	3	0.6	56	8	-0.89	48.07	0	2	6
R101	3	1	90	8	-0.31	98.67	0	4	4
total				64	-0.29	54.25	4	42	18

Case 3: $\delta=30$

C101	2	0.6	44	8	-0.04	30.87	0	7	1
C101	2	1	60	8	-0.15	46.21	0	7	1
C101	3	0.6	56	8	-0.13	50.06	0	6	2
C101	3	1	90	8	0.23	84.65	3	4	1
R101	2	0.6	44	8	-6.67	30.99	0	0	8
R101	2	1	60	8	-7.67	52.84	0	0	8
R101	3	0.6	56	8	-6.43	48.26	0	0	8
R101	3	1	90	8	-7.59	90.94	0	0	8
total				64	-3.56	54.35	3	24	37

Case 4: $\delta=20$

C101	2	0.6	44	8	-4.32	30.41	0	0	8
C101	2	1	60	8	-3.44	41.52	0	0	8
C101	3	0.6	56	8	-3.80	43.25	0	0	8
C101	3	1	90	8	-2.76	80.34	0	0	8
R101	2	0.6	44	8	-14.94	28.73	0	0	8
R101	2	1	60	8	-14.98	40.78	0	0	8
R101	3	0.6	56	8	-14.90	43.97	0	0	8
R101	3	1	90	8	-14.10	88.89	0	0	8
total				64	-9.15	49.74	0	0	64

Case 5: $\delta=10$

C101	2	0.6	44	8	-20.81	33.21	0	0	8
C101	2	1	60	8	-17.86	44.37	0	0	8
C101	3	0.6	56	8	-18.68	55.91	0	0	8
C101	3	1	90	8	-16.96	96.62	0	0	8
R101	2	0.6	44	8	-21.56	30.85	0	0	8
R101	2	1	60	8	-18.42	42.43	0	0	8
R101	3	0.6	56	8	-21.55	44.18	0	0	8
R101	3	1	90	8	-18.02	77.60	0	0	8
total				64	-19.23	53.15	0	0	64

Case 6: $\delta=5$

C101	2	0.6	44	8	-15.40	30.55	0	0	8
C101	2	1	60	8	-14.01	44.46	0	0	8
C101	3	0.6	56	8	-14.35	53.59	0	0	8
C101	3	1	90	8	-13.00	96.31	0	0	8
R101	2	0.6	44	8	-20.46	31.24	0	0	8
R101	2	1	60	8	-15.97	41.87	0	0	8
R101	3	0.6	56	8	-20.54	47.92	0	0	8
R101	3	1	90	8	-16.92	74.95	0	0	8
total				64	-16.33	52.61	0	0	64

Table 19: Results for SPD with different values of δ in the instances.